



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

VITOR RODRIGUES SANTOS

**CARRETILHA INSTRUMENTADA PARA PESCA ESPORTIVA PESADA E
DE ESPERA**

Orientador: Prof. MSc. Luciano Henrique Duque

Brasília
Julho, 2016.

CARRETILHA INSTRUMENTADA PARA PESCA ESPORTIVA PESADA E DE ESPERA

Trabalho de Conclusão de Curso apresentado à Banca examinadora do curso de Engenharia da Computação da FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas – Centro Universitário de Brasília como requisito para obtenção do título de Engenheiro da Computação.

Orientador: Prof. MSc. Luciano Henrique Duque

Brasília

Julho, 2016.

CARRETILHA INSTRUMENTADA PARA PESCA ESPORTIVA PESADA E DE ESPERA

Trabalho de Conclusão de Curso apresentado à Banca examinadora do curso de Engenharia da Computação da FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas – Centro Universitário de Brasília como requisito para obtenção do título de Engenheiro da Computação.

Orientador: Prof. MSc. Luciano Henrique Duque

BANCA EXAMINADORA

Prof. Dr. Abiézer Amarília Fernandes
Coordenador do Curso

Prof. MSc. Luciano Henrique Duque
Orientador

Prof. MSc. Luís Cláudio Lopes de Araújo
Prof.

Prof. MSc. David de Almeida Fiorillo
Prof.

Brasília

Julho, 2016.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me dar forças para acordar todos os dias com vontade de buscar a realização dos meus sonhos e por me mostrar que as barreiras que a vida nos impõe até o caminho do sucesso são apenas fontes de aprendizado.

Agradeço a minha família pelo apoio nos tempos difíceis e por compartilhar comigo momentos memoráveis. Em especial, agradeço a meu pai Reginaldo Pereira dos Santos Filho e a minha mãe Virginia do Socorro Rodrigues Costa por acreditarem no meu potencial, serem exemplo de pessoas de bom caráter e educação, e por sempre investirem sem medida nos meus estudos. Não menos importante, agradeço ao meu irmão Eduardo Rodrigues Santos, a minha irmã Milena Rodrigues Santos e a minha avó Iolanda Rodrigues Costa por estarem sempre comigo nos meus momentos felizes dos últimos anos.

Agradeço a minha namorada Larissa Alves de Oliveira por estar sempre ao meu lado me apoiando nas minhas decisões. Também, pela empatia, amor, carinho que me motivaram em todas as fases deste projeto.

Agradeço aos meus amigos que juntos estamos a quase cinco anos empenhados para nos tornarmos engenheiros de computação. No início como colegas e hoje se tornaram amigos que pretendo levar para toda a vida. Em especial agradeço ao Hugo Molina, Diego Jardim e Octávio Uchoa.

Por fim, agradeço ao Professor Mestre Luciano Henrique Duque pela ótima orientação e paciência que resultou neste trabalho. Também agradeço ao professor Doutor Abiézer Amarília Fernandes, coordenador do curso de engenharia de computação, pelas ótimas dicas e por ampliar meu campo de visão como engenheiro.

Obrigado a todos!

Vitor Rodrigues Santos

“Perguntas de alto nível criam uma vida de alto nível. Pessoas bem-sucedidas fazem melhores perguntas, e como resultado, obtêm melhores respostas. ”

Anthony Robbins

RESUMO

A pesca esportiva é uma modalidade de pesca praticada por milhões de pessoas no mundo inteiro e vem crescendo a cada dia ganhando lugar em setores como o turístico e econômico. Os equipamentos utilizados nessa modalidade estão em constante desenvolvimento, despertando o interesse dos pescadores. O projeto propõe o desenvolvimento de um protótipo de carretilha instrumentada, focado na pesca esportiva pesada de espera, com hardware e software que apresentam ao usuário funcionalidades que o ajudarão em suas pescarias, sendo elas: medição da quantidade de linha no carretel e da linha solta; alarme para alertar o usuário quando algum peixe puxar a linha; medição da profundidade do local pescado; cadastramento de três bitolas de linha; e ajuste da linha perdida durante a pescaria. Com o auxílio de um Arduino e nele acoplado uma *Shield LCD Keypad*, o usuário poderá comandar essas funcionalidades por meio de botões e ter as informações necessárias para capturar o troféu da sua pescaria. Essa solução visa aumentar a comodidade e eficácia na captura dos peixes sem perder a esportividade.

Palavras chaves: Arduino; pesca esportiva pesada de espera; carretilha instrumentada.

ABSTRACT

Sport fishing is a fishing modality practiced by millions of people worldwide and is growing every day gaining place in sectors such as tourism and economic. The equipment used in this mode are in constantly developing, attracting the interest of fishermen. The project proposes the development of a prototype instrumented reel, focused on heavy sport fishing waiting, hardware and software presents the user with features that will help in their fisheries, which are: line amount measurement on the spool and the loose line; alarm to alert the user when a fish pulling the line; fish depth measurement site; registration of three line gauges; and adjustment of the lost line while fishing. With the help of an arduino and it coupled one Shield LCD Keypad, the user can control these features using buttons and have the information needed to capture the trophy of their fishery. This solution aims to increase the convenience and effectiveness in fish capture without losing sportiness.

Key words: arduino; heavy sportfishing waiting; instrumented reel.

LISTA DE FIGURAS

Figura 2.1 - Carretilha de pesca antiga	23
Figura 2.2- Principais componentes da carretilha Marine Sports Caster 400	25
Figura 2.3 - Arduino UNO R3.....	27
Figura 2.4 - Componentes arduino UNO R3.....	28
Figura 2.5 - Pinagem microcontrolador ATmega328P	29
Figura 2.6 – Interface IDE.....	30
Figura 2.7 – Experimento efeito hall	31
Figura 2.8 – Sensor de efeito hall A3144	32
Figura 2.9 – Ligação interna sensor de efeito hall A3144.....	33
Figura 2.10 –Funcionamento da chave do sensor de efeito hall.....	34
Figura 2.11 –Micro chave KW10-B	35
Figura 2.12 – Componentes micro chave KW10-B	35
Figura 3.1 – Diagrama de blocos do desenvolvimento do projeto	37
Figura 3.2 – Visão geral das funcionalidades da carretilha instrumentada	39
Figura 3.3 – Processo medição	40
Figura 3.4 – Processo cadastramento de linha.....	41
Figura 3.5 – Processo medição de profundidade	41
Figura 3.6 – Processo ajuste	42
Figura 3.7 – Processo alarme.....	43
Figura 3.8 – Visão geral dos componentes do projeto	43
Figura 3.9 – Ligações shield LCD keypad e arduino UNO R3.....	45
Figura 3.10 – Associação de resistores para acionar os botões	46
Figura 3.11 – Pinos livres para uso shield LCD keypad 1602	46
Figura 3.12 – Carretel com ímã e sensores.....	47
Figura 3.13 – Conexões sensores efeito hall e arduino UNO R3	48
Figura 3.14 – Menu funções	49
Figura 3.15 – Cadastramento.....	50
Figura 3.16 – Fluxograma cadastramento	51
Figura 3.17 – Implementação dos quadros no programa pela função MedicaoVolta	54
Figura 3.18 – Fluxograma medição	55

Figura 3.19 – Funcionalidade ajuste visor LCD.....	57
Figura 3.20 – Marcações na linha do carretel.....	58
Figura 3.21 – Medição da linha com paquímetro	59
Figura 3.22 – Circuito montado na prothoboard	60
Figura 3.23 – Sensores de efeito hall.....	61
Figura 3.24 – Suporte para os sensores e disposição do imã.....	62
Figura 3.25 – Adaptação no chassi da carretilha	62
Figura 3.26 – Projeto placa de circuito impresso ISIS	63
Figura 3.27 – Projeto placa de circuito impresso ARES	64
Figura 3.28 – Placa de circuito impresso.....	64
Figura 3.29 – Disposição dos componentes dentro da caixa de comando	65
Figura 3.30 – Caixa de comando	65
Figura 3.31 - Interior do alarme de pesca.....	66
Figura 3.32 - Versão final do protótipo	67
Figura 4.1 – Primeiro cenário de teste	69
Figura 4.2 – Segundo cenário de teste	72
Figura 4.3 – Valores aferidos 70m na carretilha e odômetro mecânico linha 0,45	74

LISTA DE QUADROS

Quadro 3.1 - Pinos de conexão arduino e LCD keypad, e suas funções	44
Quadro 3.2 - Medição linha 0,50	53
Quadro 4.1 – Resultado primeiro cenário de teste.....	70
Quadro 4.2 – Erro do cadastramento da linha 0,40	73
Quadro 4.3 – Erro do cadastramento da linha 0,45	73
Quadro 4.4 – Erro do cadastramento da linha 0,50	74
Quadro 4.5 – Erro da funcionalidade medição linha 0,40 mm.....	76
Quadro 4.6 – Erro da funcionalidade medição linha 0,45 mm.....	77
Quadro 4.7 – Erro da funcionalidade medição linha 0,50 mm.....	78
Quadro 4.8 – Custo do projeto	789

LISTA DE SIGLAS

A	<i>Ampere</i>
AC	<i>Alternate Current</i>
AVR	<i>Advanced Virtual RISC ou Alf and Vergad RISC</i>
CPU	<i>Central Processing Unit</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
GND	<i>Ground</i>
IDE	<i>Integrated Development Environment</i>
LCD	<i>Liquid Crystal Display</i>
NC	<i>Normaly Closed</i>
NO	<i>Normaly open</i>
PWM	<i>Pulse width modulation</i>
RAM	<i>Random Access Memory</i>
RISC	<i>Reduced Instruction Set Computer</i>
USB	<i>Universal Serial Bus</i>
V	<i>Volts</i>
VCC	<i>Collector supply voltage</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivo Geral	16
1.2	Objetivos Específicos	16
1.3	Motivação	17
1.4	Metodologia.....	17
1.5	Resultados Esperados	19
1.6	Estrutura do Trabalho	19
2	REFERENCIAL TEÓRICO.....	21
2.1	Pesca Esportiva.....	21
2.2	Carretilhas de Perfil Alto	23
2.3	Arduino Uno R3	26
2.3.1	Hardware.....	26
2.3.2	Software	30
2.4	Sensores.....	31
2.4.1	Sensor de Efeito Hall	31
2.4.2	Micro Chave KW10-B	34
3	DESENVOLVIMENTO DO TRABALHO PROPOSTO.....	37
3.1	Apresentação da Proposta.....	38
3.2	Componentes do Projeto.....	43
3.2.1	Arduino Uno R3 e Shield LCD Keypad 1602.....	44
3.2.2	Carretilha e Sensor de Efeito Hall A3144	47
3.2.3	Sensor de Efeito Hall A3144 e Arduino Uno R3	48
3.3	Software do Projeto	49
3.3.1	Cadastramento de Nova Linha	50
3.3.2	Medição.....	54
3.3.3	Medições da Profundidade	57

3.4	Medida do Valor das Voltas	57
3.5	Montagem do Protótipo	59
3.5.1	Montagem do Circuito na Protoboard	60
3.5.2	Adaptação no Chassi da Carretilha	61
3.5.3	Placa de Circuito Impresso.....	63
3.5.4	Montagem Final do Protótipo na Caixa e Fixação na Vara.....	64
4	TESTES E RESULTADOS	68
4.1	Primeiro Cenário.....	68
4.1.1	Descrição.....	68
4.1.2	Pré-requisitos	69
4.1.3	Resultados	69
4.2	Segundo Cenário.....	71
4.2.1	Descrição.....	71
4.2.2	Pré-requisitos	71
4.2.3	Resultados	72
4.3	Terceiro Cenário	75
4.3.1	Descrição.....	75
4.3.2	Pré-requisitos	75
4.3.3	Resultados	76
5	CONSIDERAÇÕES FINAIS	80
5.1	Conclusão	80
5.2	Propostas de Trabalhos Futuros	81
	REFERÊNCIAS BIBLIOGRÁFICAS.....	82
	APÊNCIDE A- CÓDIGO FONTE DA CARRETILHA INSTRUMENTADA NA PLATAFORMA ARDUINO	85
	APÊNDICE B- QUADROS DE MEDIÇÃO DAS BITOLAS DE LINHA 0.40, 0.45 E 0.50 ...	113

1 INTRODUÇÃO

Engrenada pela necessidade do homem, a tecnologia, a cada dia, vai se desenvolvendo mais com os objetivos de: resolver problemas; aumentar a comodidade e qualidade da vida do homem; aumentar eficiência e eficácia na produção de um item; desenvolvimento de novos produtos; unir o homem de diversos lugares do planeta; dentre outros.

Vários são os setores que a tecnologia atua e a pesca é um deles. Antes a pescaria feita com lanças, arco e flecha, anzóis simples e hoje com carretilhas e molinetes que possuem diversas funcionalidades e mais de 50 peças em sua construção.

Desde as primeiras civilizações humanas, a pesca vem fazendo parte da vida do ser humano como uma das principais fontes de alimento. “No Brasil, a geografia generosa de grandes rios e afluentes sempre favoreceu a atividade, de modo que mesmo antes do “descobrimento” a pesca já havia se estabelecido entre os indígenas. Quando os portugueses aqui atracaram, encontraram tribos nativas com seus métodos próprios para a construção de canoas e utensílios para a captura de peixes.” (IBAMA, 2008)

O Decreto Lei 221 de 28 de fevereiro de 1967 define a pesca como o ato de capturar ou extrair animais ou vegetais que tenham na água o seu meio de vida classificando-a em pesca comercial e não comercial. A pesca comercial pode ser artesanal praticada por pescador profissional autônomo, ou em regime familiar e a industrial realizada por pessoas físicas ou jurídicas em que pescadores profissionais empregados ou em regime de parceria se utilizam de embarcações de pequeno, médio e grande porte constituindo uma atividade empresarial, sendo que ambas visam a mercantilização do pescado. A pesca não comercial é composta pela pesca de subsistência, na qual seus praticantes visam o consumo doméstico, a pesca científica visando o desenvolvimento de pesquisas e por fim, a pesca amadora que é praticada por brasileiro ou estrangeiro com a finalidade de lazer ou o desporto (NOGUEIRA; GHEDIN, 2010). Assim, a pesca esportiva ou de lazer é uma modalidade da pesca amadora, também chamada de pesca esportiva, e define-se como a captura de peixes visando uma atividade de lazer (PITCHER, 1999) que não se destinam para a alimentação (ARLINGHAUS, 2009).

A pesca esportiva é uma importante atividade na economia de diversos países, gerando de forma direta ou indireta bilhões de dólares em renda e impulsiona um segmento industrial desenvolvido (DONALDSON, 2011; STOECKL; GREINER; MAYOCCHI, 2006).

Esse tipo de pescaria em decorrência da sua baixa captura de exemplares por esforço de pesca é apontada como menos prejudicial para os ecossistemas aquáticos do que a pesca comercial que se caracteriza por uma grande captura aliado a um baixo esforço de pesca, no entanto, a atividade pode produzir impactos diretos e indiretos aos recursos naturais envolvidos (COOKE; COWX, 2006; ARLINGHAUS, 2012). O excesso de capturas superiores aos níveis sustentáveis, ou sobrepesca, afeta: o tamanho e as abundâncias das populações (BARTHOLOMEW; BOHNSACK, 2005).

Um tipo de pescaria realizada dentro da pesca esportiva é a pesca pesada de espera. Em outros tipos o pescador vai atrás do peixe, nesse, a lógica é inversa, a isca é armada e acontece a espera até que o peixe a ataque. Em geral, esse tipo de pescaria é feito com equipamento mais robustos, que possuem maior capacidade de linha de libragens mais alta e maior frenagem no carretel, pelo motivo dos peixes serem maiores.

A pesca de espera pode ser realizada em qualquer período do dia. Em rios ou pesqueiros (lagos artificiais), principalmente à noite, pois peixes de couro são alvo desse estilo de pesca devido ao seu hábito de alimentação nesse período. (OLO, 2014)

A pesca pesada de espera é uma modalidade que precisa de bastante preparação, paciência, atenção e equipamento adequado. Preparação, pois é importante obter, por exemplo, informações da profundidade para saber onde arremessar com objetivo de encontrar locais com mais peixes. Como o peixe pode atacar a isca em poucos minutos de espera ou em horas, é comum que o pescador faça outras atividades e não esteja atento quando o peixe começar a puxar a linha. É aqui que um dispositivo que avise quando a linha começar a correr faz-se necessário para que a fígada não seja perdida e horas e horas de espera sejam jogadas fora.

Se o equipamento utilizado não estiver regulado corretamente para a força imposta pelo peixe, pode ser que ele consiga em pouco tempo levar toda a linha do carretel. Para ajudar o pescador a não perder o peixe por esse motivo, como uma das funcionalidades desse projeto, o usuário poderá gerenciar a quantidade de linha do carretel, em metros, que serão mostrados em um visor LCD.

Hoje, no mercado, existem equipamentos de pesca denominadas carretilhas elétricas, que possuem a maioria das funcionalidades propostas nesse projeto e muitas outras, como por exemplo: recolhimento automático da linha. Esse equipamento é focado em pesca de grandes profundidades (mais de 100 metros) que acontecem essencialmente no mar e possui um custo muito elevado. Por possuir essas características, não é utilizada na pesca pesada de espera em água doce, sendo a proposta desse projeto trazer funcionalidades adaptadas para esse tipo de pesca, que aumente a eficácia na captura dos peixes de rios e lagos.

Nesse cenário o projeto proposto apresenta um protótipo focado na pesca esportiva pesada de espera, tanto em rio, quanto em lagos de água doce, objetivando minimizar as eventuais situações que possam fazer com que o pescador perca o peixe. Disponibilizando ao pescador uma automatização com funções que lhe permitirão: saber a profundidade do local pescado; saber por meio de um visor de LCD a quantidade de linha, em metros, que contém no carretel e quantidade de linha que foi descarregada; ser avisado que a linha está acabando, para que ele tome alguma atitude a fim de não perder o peixe; e ser avisado, por meio de um alarme, quando o peixe está puxando a linha. Tendo sempre o cuidado para que o pescador tenha uma maior comodidade sem tirar a esportividade.

1.1 Objetivo Geral

O objetivo geral do projeto é desenvolver um protótipo de dispositivo para pesca esportiva pesada de espera, dotados de funcionalidades básicas que irão ajudar os pescadores dessa modalidade na captura dos peixes.

1.2 Objetivos Específicos

- Desenvolver uma placa de circuito impresso com os sensores para leitura da quantidade de linha.
- Utilizar uma *shield* de *display* LCD com botões em conjunto com arduíno UNO R3 para comandar as funcionalidades do protótipo.
- Elaborar um programa em C/C++ que contemplará as funções da carretilha.

- Modificar o Chassi da carretilha para acomodar os sensores.
- Fazer testes no *Hardware* e no programa da carretilha para obter resultados qualitativos e quantitativos.

1.3 Motivação

A motivação deste trabalho é o auxílio que as funcionalidades da carretilha instrumentada irão gerar ao pescador, levando a uma maior eficácia na captura dos peixes. Além disso, reunir funções encontradas no mercado em equipamentos separados, em um único dispositivo.

Os equipamentos que possuem essas funcionalidades agrupadas são as carretilhas elétricas, possuindo um custo bastante elevado e possuem um foco diferente da pescaria pesada de espera em lagos e rios. É inspirado nessa falta de equipamento automatizado para esse tipo de pescaria que esse projeto é proposto.

1.4 Metodologia

Com a carretilha instrumentada de pesca pesada de espera, os pescadores não precisarão ter preocupações em alguns fatores que antes a desatenção deles poderia ser o limiar entre o sucesso e o fracasso na pescaria. Para a implementação do projeto proposto, será utilizado dois sensores de efeito Hall para colher dados e a plataforma de desenvolvimento arduino Uno R3 fica a cargo do processamento dos mesmos, sendo programado na sua linguagem própria de programação que é baseada nas linguagens C e C++. O arduino possui a vantagem de facilitar a prototipagem, possuir uma vasta quantidade de componentes que nele se encaixa, ter uma IDE (ambiente de desenvolvimento) multi-plataforma que pode ser instalado em diferentes sistemas operacionais e de possuir uma vasta quantidade de bibliotecas *online*.

É necessário seguir a seguinte metodologia, com 6 etapas, para que seja alcançado os objetivos específicos do projeto:

Primeira etapa: compreende a pesquisa bibliográfica e estudo dos componentes que envolvem o projeto:

- Arduino e componentes elétricos;
- Programação no IDE do arduino baseado nas linguagem C e C++;
- Equipamentos e funcionalidades encontradas no mercado, para efeito de comparação;
- Sensor de efeito *Hall*.

Segunda etapa: desenvolvimento do *hardware*, envolvendo a montagem do circuito elétrico utilizando, inicialmente, uma placa de ensaio (*protoboard*). Essa etapa visa, também, identificar quais são os componentes eletrônicos mais adequados para o projeto.

Terceira etapa: desenvolver um *software* da carretilha na linguagem de programação própria do arduino, baseada em C/C++, contemplando um menu que será controlado e mostrado por um visor LCD com botões, dando acesso a todas as suas funcionalidades:

- Cadastramento de linha;
- Medição da profundidade do local pescado;
- Alarme;
- Medição da quantidade de linha presente e solta do carretel;
- Ajuste do total de linha;

Quarta etapa: essa etapa consiste na transposição do circuito montado na placa de ensaio (*protoboard*) para uma placa de circuito impresso. Tendo em vista a restrição de espaço, para montar uma placa de circuito impresso que não fique grande a ponto de ter que fazer muitas alterações no chassi, podendo prejudicar na ergonomia.

Quinta etapa: adaptação do chassi da carretilha para postar os sensores que fazem a leitura da quantidade de voltas dada pelo carretel da carretilha, montagem e fixação na vara do alarme e de uma caixa de plástico protetora, onde ficará boa parte do hardware do projeto.

Sexta etapa: destinada a testes tanto na parte de *hardware* quanto *software*, pequenos ajustes após a adaptação no chassi e fixação de todo o equipamento na vara. Os testes estão direcionados a:

- Erro da medição de linha presente e solta do carretel;
- Erro no cadastramento da quantidade de linha no carretel;
- Testes nos sensores de efeito Hall que serão usados na medição da linha.

1.5 Resultados Esperados

Após implementação do protótipo é esperado que tenhamos uma integração das funcionalidades em um único aparelho de comando, que o usuário terá acesso por meio de um Menu, dentre elas: alarme sonoro que será disparado quando houver uma tração na linha; quantidade de linha descarregada e presente no carretel; cadastramento de três bitolas de linha na carretilha e medição da profundidade do local pescado.

É esperado, também que a leitura de medição da quantidade de linha do carretel e linha liberada, seja mostrada em um visor LCD.

Como a esportividade é um fator importante na pesca esportiva pesada de espera, espera-se que o projeto atenda às necessidades dos pescadores desse estilo, tendo cuidado para não introduzir funções em excesso que comprometam a esportividade.

1.6 Estrutura do Trabalho

O trabalho contém uma estrutura dividida em capítulos e subtítulos:

O primeiro capítulo se refere à introdução do tema que será abordado no projeto, destacando a importância de ter as funcionalidades oferecidas no trabalho; bem como a apresentação do objetivo geral e objetivos específicos; a metodologia, que se refere a uma sequência de passos para chegar a estrutura final do projeto; a motivação que levou a confecção deste trabalho e os resultados esperados.

O segundo capítulo é destinado ao referencial teórico, onde será apresentada a proposta do projeto citando as características do arduino e dos componentes eletrônicos que serão utilizados.

O terceiro capítulo mostrará o desenvolvimento do projeto, descrevendo as técnicas, metodologias e ferramentas utilizadas.

O quarto capítulo é composto dos testes realizados no protótipo, descrevendo os resultados e as variáveis que os influenciaram.

O quinto capítulo apresenta a conclusão do trabalho e sugestões de trabalhos futuros com a experiência obtida.

2 REFERENCIAL TEÓRICO

Esse capítulo é destinado a fazer uma explanação dos assuntos que integrarão a proposta de solução do projeto, sendo dividido em 4 seções. Será tratado, inicialmente, uma seção sobre a pesca esportiva. A segunda seção é referente às carretilhas de perfil alto, explicando seus componentes e sua finalidade na pesca. Após, trataremos da plataforma de desenvolvimento arduino, o dividindo em *hardware* e *software*: na primeira parte falaremos das especificações técnicas da versão do arduino utilizada nesse projeto e na segunda, do ambiente IDE (ambiente de desenvolvimento). Finalizando, na última seções será explicado sobre o funcionamento dos sensores utilizados.

2.1 Pesca Esportiva

A pesca é considerada como uma atividade de extração de organismos no ambiente aquático que vem sendo praticada desde os primórdios da sociedade. Era uma das principais fontes de alimento do ser humano, para sua subsistência ou como complemento da sua alimentação sem o intuito de venda ou troca. Com o passar dos anos e depois com o surgimento de comunidades, vilas e cidades o homem iniciou então a produção de alimentos para a comercialização e para a geração de riquezas. (GARCIA, 2013)

Segundo o art. 8º da lei nº 11.959 de 2009, a pesca é classificada em:

I – Comercial:

- a) artesanal: quando praticada diretamente por pescador profissional, de forma autônoma ou em regime de economia familiar, com meios de produção próprios ou mediante contrato de parceria, desembarcado, podendo utilizar embarcações de pequeno porte;
- b) industrial: quando praticada por pessoa física ou jurídica e envolver pescadores profissionais, empregados ou em regime de parceria por cotas-partes, utilizando embarcações de pequeno, médio ou grande porte, com finalidade comercial;

II – Não comercial:

- a) científica: quando praticada por pessoa física ou jurídica, com a finalidade de pesquisa científica;
- b) subsistência: quando praticada com fins de consumo doméstico ou escambo sem fins de lucro e utilizando petrechos previstos em legislação específica.
- c) amadora: quando praticada por brasileiro ou estrangeiro, com equipamentos ou petrechos previstos em legislação específica, tendo por finalidade o lazer ou o desporto;

Também chamada de pesque e solte ou pesca amadora, a pesca esportiva é uma modalidade em que os peixes são soltos logo após a sua captura, buscando a esportividade e também a conscientização ambiental.

O produto da pesca amadora pode ser utilizado com fins de consumo próprio, ornamentação e obtenção de isca vivas para pesque e solte. As atividades relacionadas a pesca esportiva podem ter finalidade econômica, sendo proibido a comercialização dos peixes pescados. (MPA/MMA, 2012)

A pesca esportiva é procurada por puro prazer ou *hobbie*, mas também com intuito de fugir da monotonia e estresse do cotidiano das grandes cidades. O esporte tem crescido bastante nas últimas décadas, podemos ver esse cenário pela quantidade de programas de televisão, “pesque e pagues”, campeonatos e pacotes turísticos voltados para este tipo de pesca. (SILVEIRA A. , 2013)

Com o crescimento do esporte, antigas modalidades vão sendo aprimoradas e novas vão surgindo. A pesca de espera, definida como linhada ou vara iscada que se coloca para que o peixe se fisque sozinho (FABICHAK, 1995), é uma modalidade onde o pescador mexe no equipamento poucas vezes até que o peixe seja fispado. Apesar de ser utilizada a bastante tempo, hoje, a pesca de espera vem sendo utilizada pelos pescadores enquanto praticam outras modalidades com mais ação focando em peixes de couro, tanto em rio, quanto em pesqueiros de pesque e solte.

Sendo o foco desse trabalho construir um protótipo de carretilha (de perfil redondo ou alto) para pesca esportiva pesada de espera, podemos definir pesada como uso de equipamentos (vara e carretilha) mais fortes e maior dimensionados se comparado com outras modalidades. Geralmente, são usadas carretilhas de perfil alto para esse tipo de pesca.

2.2 Carretilhas de Perfil Alto

As primeiras carretilhas da história foram ilustradas na China, no ano de 1195 (BENSUSAN, 2008). Eram bastante simples possuindo apenas um sistema que enrolava a linha sem um guia de linha. A figura 2.1 exemplifica o sistema utilizado nas primeiras carretilhas de pesca.

Figura 2.1 - Carretilha de pesca antiga



Fonte: disponível em http://www.ciadapesca-rs.com.br/conteudo/0,6235_historia-das-carretilhas-de-pesca

A carretilha surgiu da necessidade de o pescador ter mais linha à sua disposição, para um arremesso mais longo ou para o caso de fregar um peixe maior. A primeira menção a carretilha aparece na obra de Thomas Baker, *The art of angling* (1651; A arte da pesca desportiva). (TANIKAWA, 2016)

As primeiras carretilhas que se têm notícia, eram bastante rudimentares, tinham o seu corpo preso a uma vara e uma manivela para recolher a linha. Não possuíam nenhum mecanismo de frenagem e sua ação era simples, ou seja, cada volta que se dava na manivela correspondia a mesma proporção do carretel. Com a modernidade e a evolução industrial, surgiram carretilhas melhores, mais robustas e com mais funcionalidades, temos como exemplo: as carretilhas automáticas, que recolhem a linha automaticamente, usadas bastante na pesca marítima, as carretilhas de *fly* e as carretilhas multiplicadoras. Essas últimas possuíam um novo sistema que aperfeiçoou a ação simples das anteriores, um sistema multiplicador que a cada volta dada na manivela, o carretel gira numa proporção maior, fazendo com que os

pescadores consigam recolher a linha mais rapidamente, poupando esforços na sua pescaria. (FUKUMOTO, 1994)

Alguns fabricantes de carretilha foram responsáveis pelo grande avanço tecnológico que tivemos no século 19 no ramo, entre eles: Jonathan F. Meek, John Hardman e Benjamin Milan. O sistema de guia de linha, utilizado até hoje, foi desenvolvido pela primeira vez na empresa de Wisconsin Wheeler & McGregor (BENSUSAN, 2008), um sistema que acomoda a linha de forma uniforme no carretel.

O desenvolvimento das carretilhas de pesca é constante, hoje, encontramos no mercado carretilhas diferentes de acordo com a pesca alvo. Os principais tipos de carretilhas são as carretilhas de perfil alto (redondo) e as carretilhas de perfil baixo (anatômico). Ambos os tipos possuem os mesmos componentes, mas trazem formatos diferentes. A carretilha de perfil alto possui um aspecto cilíndrico, consegue armazenar uma maior quantidade de linha, é mais comprida e pesada, e tem uma maior força de tração. Enquanto isso, a de perfil baixo é mais achatada, leve e prática, apresenta mais recursos e tem maior controle e precisão no arremesso. (CREMOSI, 2015)

Podemos dizer que as carretilhas são divididas em vários níveis, independentemente do tipo, que vão de leve até extrapesada. Cada uma é utilizada para determinado objetivo, dependendo do tipo de peixe que se espera capturar ou da bitola da linha de pesca (CREMOSI, 2015):

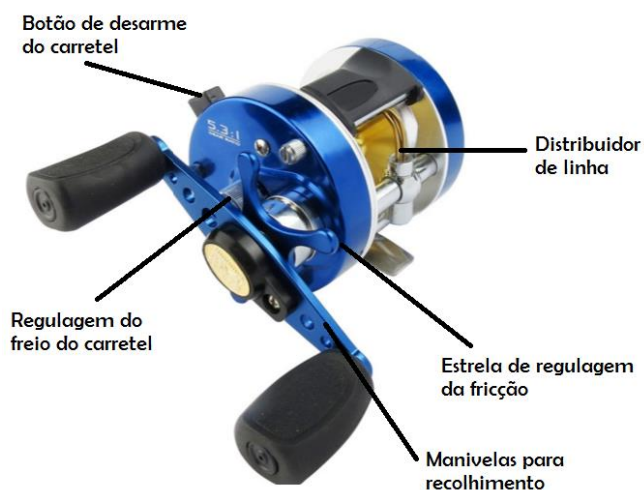
- As carretilhas que possuem bitola de linha de 0,14 a 0,20mm são consideradas leves e geralmente são usadas para pesca de peixes de pequeno porte. (PESCA AMADORA, 2016);
- As carretilhas que se encontram entre as bitolas de 0,23 a 0,37mm estão no nível médio, são carretilhas bastante utilizadas em pesqueiros e pesque e pague para peixes de médio porte. (PESCA AMADORA, 2016);
- As carretilhas pesadas são aquelas que usam bitolas de linha entre 0,40 e 0,62 mm e são focadas na pesca de peixes maiores, geralmente os de couro. (PESCA AMADORA, 2016);
- Já as carretilhas extrapesadas são focadas na pescaria marítima de grande, envolvendo grandes profundidades, com bitolas de linha acima de 0,62 mm (PESCA AMADORA, 2016).

O equipamento que será usado neste projeto é uma carretilha da marca *Marine Sports Caster 400* com manivela do lado direito. Essa carretilha é considerada se localiza no nível de um equipamento pesado, pois as bitolas de linha usadas estão entre 0,40 e 0,62 mm. Este equipamento possui as seguintes especificações (MGPESCA, 2016):

- Freio: centrífugo:
- Peso: 450 gramas
- Manivela direita: 400
- Manivel esquerda: 400L
- *Drag* (fricção): 9 lb (4,5 kg)
- Relação de recolhimento: 5.3:1
- Cap. de linha: 0,40 mm – 190 m
- Rolamentos: 2 de esferas + 1 de roletes

A figura 2.2 mostra os principais componentes da *Caster 400* e logo após, a explicação dos componentes:

Figura 2.2- Principais componentes da carretilha Marine Sports Caster 400



Fonte: Adaptada de <http://www.pescaeturismo.com.br/pescaria/carretilhas-conheca-e-arremesse-sem-cabeleiras/>

- **Fricção:** é um dispositivo regulado por meio da estrela ilustrada na figura 2.2 que permite que o usuário escolha qual pressão desejada na liberação da linha. Com a regulagem mais branda, o peixe tem facilidade de levar a linha e quanto mais vai prendendo, maior será a pressão que tem que exercer na linha para retirá-la do carretel. Essa regulagem de acordo com o tipo de peixe que se deseja capturar e o tipo de linha que está sendo usada. (DAVID, 2009)
- **Distribuidor de linha:** mecanismo que, de acordo com o giro do carretel, no recolhimento acomoda a linha de forma uniforme.
- **Regulagem do freio do carretel:** ao girar a porca, internamente um mecanismo aperta o eixo do carretel fazendo com que o carretel gire mais solto ou mais travado. É regulado de acordo com o peso da isca. (DAVID, 2009)
- **Botão de desarme do carretel:** ao pressionar o botão o carretel é liberado para girar no sentido que desenrola a linha, assim, o pescador pode efetuar o arremesso.
- **Manivela de recolhimento:** recolhe a linha quando é girada.

2.3 Arduino Uno R3

Para um melhor entendimento da plataforma arduino, dividiremos esse subcapítulo em 2 partes: hardware, incluindo seus componentes e especificações; e software, explanando sobre a interface e linguagem de programação.

2.3.1 Hardware

O arduino foi criado, na Itália, em 2005 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. (THOMSEN, 2014) É uma plataforma de prototipagem de código aberto baseado em hardware e software fácil utilização. (ARDUINO, 2015)

Desde sua criação o arduino tem sido o principal componente de vários projetos de todos os tipos que se possa imaginar. Possui uma grande comunidade mundial de estudantes, amadores e profissionais que contribuem na criação e aperfeiçoamento conhecimento acessível para qualquer pessoa que queira explorar a plataforma. (ARDUINO, 2015)

O projeto original Italiano foi refinado por pesquisadores para se tornar mais leve, mais acessível e mais barato. A plataforma, hoje, está bastante evoluída possuindo uma compatibilidade com todos os sistemas operacionais de forma integrada. Tanto no quesito hardware, quanto no quesito software a plataforma pode ser usada para criação de outras novas aplicações, o projeto por possuir código aberto pode ser distribuído para quem quiser e também ser modificado, podendo ser utilizado até mesmo para fins comerciais. (LOPES, 2013)

Inúmeros são os tipos de arduino encontrados no mercado hoje: arduino Uno, Mega, Leonardo, Due, entre outros. O que define o modelo a ser usado é a necessidade do projetista, assim, utilizaremos nesse projeto o arduino Uno R3, pelo fato de ter um tamanho pequeno e um bom processamento, se adequando as necessidades do projeto.

Esse modelo possui um bom número de portas disponíveis e uma grande compatibilidade com *Shields* disponíveis no mercado. Possui um micro controlador ATMEGA328P, 16MHz de velocidade de *clock*, 14 portas digitais, sendo que 6 delas podem ser usadas como saídas PWM(modulação de largura de pulso), e 6 portas analógicas. (THOMSEN, 2014) A figura 2.3 ilustra o arduino Uno R3.

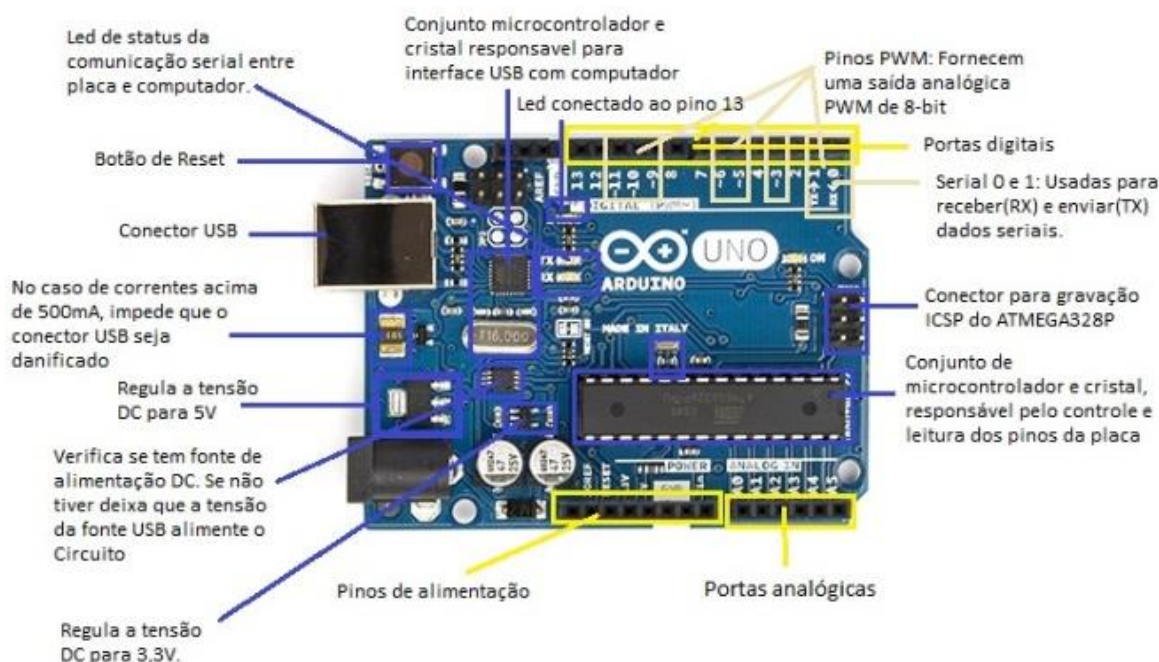
Figura 2.3 - Arduino UNO R3



Fonte: disponível em <http://blog.filipeflop.com/arduino/tipos-de-arduino-qual-comprar.html>

A figura 2.4 ilustra os componentes do arduino UNO R3, com uma breve explicação de alguns deles:

Figura 2.4 - Componentes arduino UNO R3.



Fonte: adaptado de <http://www.embarcados.com.br/arduino-uno/>

A alimentação do arduino tanto pode ser feita pelo conector USB, quando por uma fonte de alimentação externa por meio do conector *Jack*. Sendo alimentado externamente, os limites de tensão devem estar entre 6V a 20V, sendo recomendado utilizar a alimentação entre 7V e 12V, pois tensões abaixo de 7V podem trazer instabilidade ao circuito e acima de 12V o regulador de tensão pode superaquecer. Ressaltando que a tensão de funcionamento da placa do arduino Uno R3 é de 5V.

Dentre os seus pinos, como podemos observar na marcação da imagem 2.4, temos os pinos de alimentação:

- VIN – pino destinado a alimentação externa, uma alternativa para alimentar dispositivos que precisam de mais de 5V. Caso, por exemplo, a conexão externa seja alimentada por 9V o arduino utilizara 5V para a alimentação da placa, mas nesse pino estará os 9V de entrada.
- GND – terra (*ground*), pinos de referência.
- 5V – este pino fornece alimentação de 5V para circuitos externos.
- 3,3V- fornece alimentação de 3,3V para circuitos externos com corrente máxima de 50mA.

- RESET – Conectado ao RESET do ATmega328P, esse pino pode ser utilizado para dar um *reset* externo a placa do arduino.
- IOREF- Fornece tensão que pode ser adaptada entre 3,3V e 5V, de acordo com o que necessário.

O ATmega328P é o microcontrolador do arduino, que possui 2 blocos principais de memória: memória flash, que é do tipo não volátil, com 32KB para programação em aproximadamente 10.000 ciclos de leitura e escrita e a memória RAM, do tipo volátil, com 2K de memória de dados, para variáveis guardadas em registradores e variáveis temporárias externas. Contando ainda com 1K de EEPROM (memória programável apagável somente de leitura) (SILVEIRA, 2013).

Microcontrolador se refere a um sistema básico que possui uma CPU, circuitos de entrada e saída e memória. De acordo com a fabricante Atmel, o ATmega 328P possui a tecnologia AVR (*Advanced Virtual RISC* ou *Alf and Vergad RISC*) e é baseado na arquitetura RISC(*Reduced Instruction Set Computer*), ou seja, um microcontrolador com conjunto de instruções reduzido. Pode operar até 20MHz, mas o arduino Uno só opera a 16Mhz (SOUZA, 2013)

Além disso, esse microcontrolador possui 28 pinos, dos quais 23 podem ser programáveis como (*output*), para enviar dados a um circuito externo conectado ao arduino, ou podem ser programados para receber (*input*) dados de um circuito externo. Desses 23 pinos, todos podem ser utilizados com comunicação digital, porém, seis deles também podem ser configurados para receber sinal analógico. (SILVEIRA, 2013) A figura 2.5 ilustra a pinagem do ATmega328P:

Figura 2.5 - Pinagem microcontrolador ATmega328P

RESET	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	Analog5
0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	Analog4
1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	Analog3
2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	Analog2
3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	Analog1
4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	Analog0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	AREF
CRISTAL (PCINT6/XTAL1/TOSC1)	PB6	9	20	AVCC	VCC
CRISTAL (PCINT7/XTAL2/TOSC2)	PB7	10	19	PB5 (SCK/PCINT5)	13
5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	12
6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	11 (PWM)
7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	10 (PWM)
8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	9 (PWM)

Fonte: disponível em <http://blog.webtronico.com/?p=107>

2.3.2 Software

A plataforma de desenvolvimento do arduino IDE (Integrated Development Enviroment), uma aplicação multi-plataforma escrita em Java, é esquematizada para introduzir a programação a pessoas não familiarizada com o desenvolvimento de software. Possui um editor de códigos com vários recursos que ajuda quem estar programando na identificação de erros, endentação, parênteses e fácil compilação. Não sendo necessário rodar programas em ambiente de linhas de comando (HACKLAB, 2015).

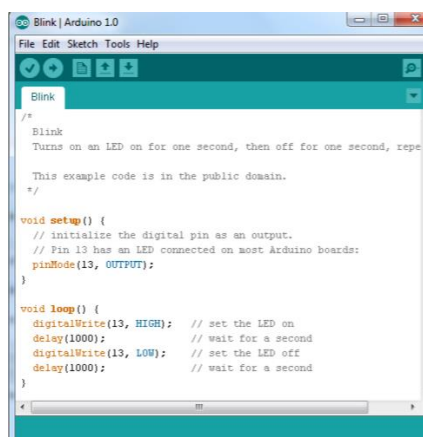
O software do arduino é livre, possuindo em seu site oficial uma grande quantidade de amostra de códigos que pode ser utilizada por qualquer pessoa que queira iniciar um projeto (O'BRIEN, 2008).

É programável tanto em C, quanto em C++, com uma interface que busca facilitar a programação. A IDE possui duas funções principais que com elas se pode fazer uma infinidade de programas:

- `setup()` – Usada para fazer configurações, inicializando os pinos de entrada e saída por exemplo;
- `loop()` – Todos os comandos que estiverem dentro dessa função serão repetidos em ordem de escrita.

A figura 2.6 mostra a interface IDE com um programa básico, como exemplo, que faz piscar um LED as duas principais funções.

Figura 2.6 – Interface IDE



Fonte: Disponível em <http://www.robotizando.com.br/pt-br/curso-de-arduino-aula-2-arquitetura-do-hardware/4>

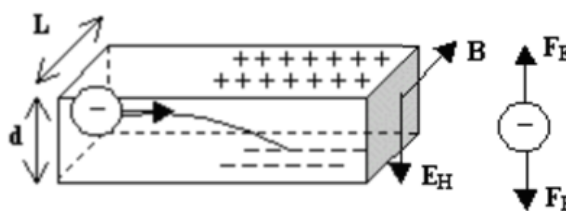
2.4 Sensores

2.4.1 Sensor de Efeito Hall

O efeito hall foi descoberto em 1879, durante uma tese de doutorado de Edwin Herbert Hall. Físico e inventor estadunidense, publicou um artigo sobre um fenômeno observado em um dos seus experimentos. Edwin observou uma diferença de potencial nos circuitos elétricos quando submetidos a um campo magnético, sendo um grande feito para física e guiando outros cientistas posteriores. (SANTOS, 2011)

Em seu experimento, o físico utilizou uma placa fina de germânio, um semicondutor conectado a um voltímetro e uma bateria. Observou pela leitura do voltímetro, a existência de uma tensão e esse efeito, a ocorrência de uma tensão no semicondutor quando foi submetido a um campo magnético foi denominado efeito Hall. (SANTOS, 2011) A figura 2.7 ilustra o experimento feito por Hall.

Figura 2.7 – Experimento efeito hall



Fonte: Disponível em http://www.if.ufrgs.br/tex/fis142/mod08/m_s03.html

Ao aplicar uma força magnética na direção horizontal, conforme ilustrado na figura 2.7, a resultante é uma força magnética na direção perpendicular ao movimento, no sentido de cima para baixo. Esta força fará com que o movimento dos elétrons seja desviado para baixo. Assim, com o tempo as cargas negativas se acumulam na parte inferior e as cargas positivas na superior. (UFRGS, 2003)

Os sensores de efeito Hall utilizam, justamente, o efeito hall para verificar se foi detectado algum campo magnético que está passando na área do sensor. Assim, podemos utilizá-lo no arduino como um dispositivo que faz a leitura se existe algum campo magnético perto dos sensores. O sensor utilizado nesse projeto é modelo A3144 que trabalha com leitura nos pinos digitais. Se não houver campo magnético, o sensor envia o sinal 1 (HIGH) para o arduino, havendo campo o sinal enviado é 0 (LOW). (THOSEN, 2014) A figura 2.8 ilustra o sensor utilizado no projeto e logo após uma descrição de principais especificações técnicas.

Figura 2.8 – Sensor de efeito hall A3144



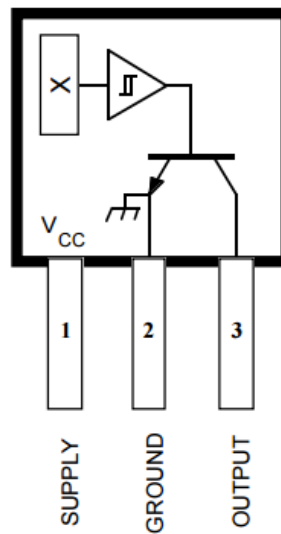
Fonte: Disponível em <http://www.autocorerobotica.com.br/a3144-sensor-de-efeito-hall>

O sensor ilustrado na figura 2.8 possui as seguintes especificações técnicas:

- Tensão de operação: 4,5 a 24 V
- Suporta temperaturas acima de 150°C
- Corrente de saída: 25mA
- Proteção contra tensão reversa

Esse sensor modelo A3144 é um circuito monolítico integrado, estável com variações de temperatura e tensão. É unipolar, ou seja, reage apenas ao campo magnético positivo ou polo norte. Quanto esse polo do ímã se aproxima do sensor, ele irá conduzir. Estando perto do sensor, ele continuará conduzindo, cessando quando o ímã for afastado, não detectando mais campo magnético (Allego MicroSystem, 2016). A figura 2.9 representa a ligação interna do sensor:

Figura 2.9 – Ligação interna sensor de efeito hall A3144



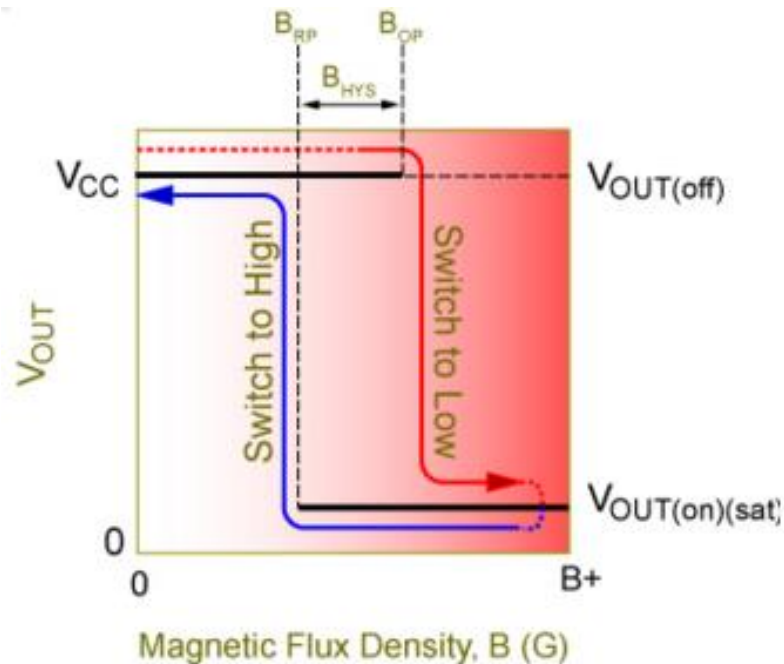
Fonte: disponível no *datasheet* <http://pdf1.alldatasheet.com/datasheet-pdf/view/55092/ALLEGRO/A3144.html>

Alguns conceitos são necessários para o entender o funcionamento do sensor (ADOLPHO, 2011):

- Bop (ponto de operação magnética): é o ponto em que o sensor hall detecta um nível de campo magnético necessário para ligar o dispositivo hall.
- Brp (ponto de liberação magnética) : ponto em que o sensor não detecta mais um campo magnético necessário para permanecer ligado e, assim, desliga.
- Bhys (histerese magnética): diferença entre |Bop e Brp|, é a diferença entre o ponto de operação e o ponto de liberação magnética. Essa função é aplicada logicamente no sensor para que sejam filtradas flutuações causadas por ruídos e vibrações mecânicas, com objetivo de ter uma saída mais limpa.

Com esses conceitos, podemos observar na figura seguinte, que quando o ponto Bop é atingido, o sensor irá conduzir trocando a chave que estava anteriormente direcionada para o terra, para a saída (output). Logicamente, o sinal que estava sendo enviado estava em HIGH (1) e logo que a chave vai para a saída, o sinal vai para LOW (0), traduzindo que o sensor está fazendo a leitura de um campo magnético. (ADOLPHO, 2011) A figura 2.10 representa esse processo:

Figura 2.10 –Funcionamento da chave do sensor de efeito hall



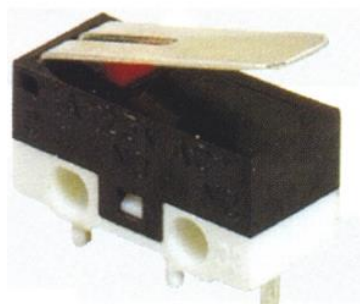
Fonte: disponível em <http://eletronicaegames.blogspot.com.br/2011/08/sensores-de-efeito-hall.html>

E quando o valor do B_{op} cai para o B_{rp} a chave volta para o valor inicial e o sensor para de conduzir. O valor lógico que estava em LOW volta para HIGH.

2.4.2 Micro Chave KW10-B

Esse sensor é um interruptor que atua no circuito, se existir uma ação mecânica pressionando o botão vermelho visto na figura 2.11, como modo *ON* ou *OFF* (liga/desliga). Esse, possui uma infinidade de aplicações, é usado, por exemplo, para detectar o fechamento de um micro-ondas ou até mesmo em mouses detectando um *click* do usuário. (SABEDORIAELETRONICA, 2014) A figura 2.11 mostra a micro chave que será utilizada neste projeto:

Figura 2.11 –Micro chave KW10-B

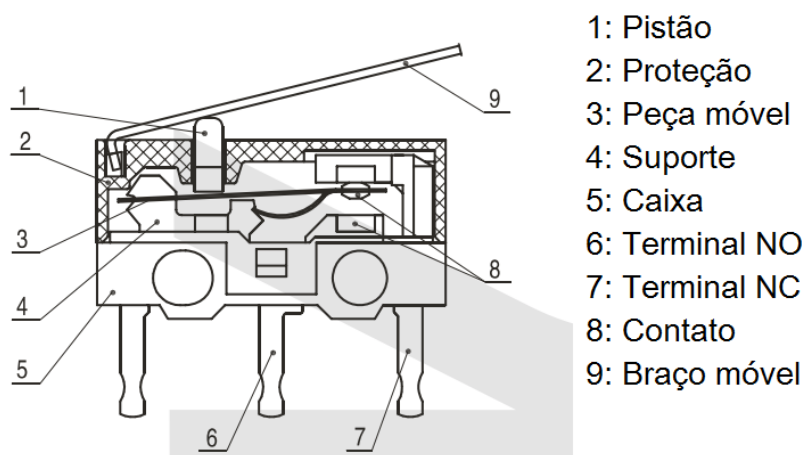


Fonte: disponível em <http://www.milcomp.com.br>

Esse modelo de micro chave possui três terminais de ligação, mas apenas dois precisam estar ligados, pois ele pode funcionar de dois modos: normalmente aberto (NO- normally open), que significa que, internamente, enquanto o botão não for pressionado ele mantém o circuito aberto sem passar nenhuma corrente e quando for pressionado fecha o circuito enviando um pulso para o circuito de controle; e o normalmente fechado(NC- normally closed), que funciona da forma inversa do primeiro. A figura 2.12 ilustra seus componentes:

Figura 2.12 – Componentes micro chave KW10-B

Desenho da configuração interna



Fonte: adaptado do datasheet disponível em http://img.ozdisan.com/ETicaret_Dosya/457408_7264032.pdf

Como podemos observar na figura 2.12, o primeiro terminal a esquerda é o de alimentação, ele obrigatoriamente precisa estar ligado, justamente com um dos outros dois terminais. Caso o 6 esteja ligado, o circuito funcionará na forma - normalmente aberto (NO) e, ao contrário, caso o 7 esteja ligado, funcionará na forma - normalmente fechado (NC).

Segundo seu *datasheet*, esse componente possui as seguintes especificações técnicas:

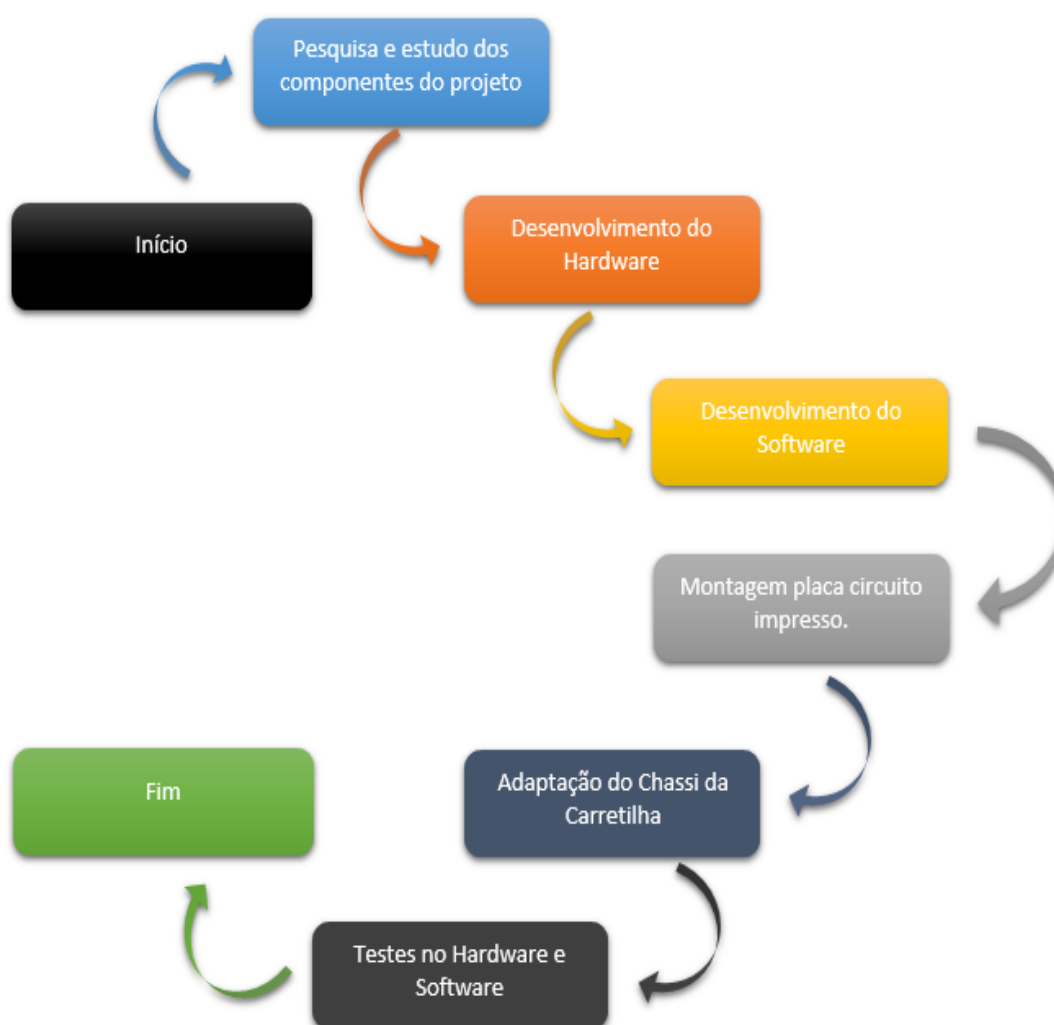
- Temperatura suportada: $-40^{\circ}\text{C} \sim +70^{\circ}\text{C}$;
- Tensão suportável: 125V AC;
- Corrente suportável: 2A;
- Velocidade de operação: 1mm - 5mm/s;
- Frequência de operação: 60 ciclos/minuto mecanicamente e 25 ciclos/minutos eletricamente;
- Peso: aproximadamente: 0,6g
- Expectativa de vida: $\geq 1.000.000$ ciclos mecanicamente e ≥ 10.000 eletricamente.

3 DESENVOLVIMENTO DO TRABALHO PROPOSTO

Este capítulo destina-se ao desenvolvimento do trabalho, que é dividido em etapas. Primeiramente é apresentado um diagrama de blocos que ilustra todas as etapas referentes ao desenvolvimento do projeto e depois, cada uma das etapas será aprofundada e exploradas.

O desenvolvimento do projeto é ilustrado, em uma visão geral, pela figura 3.1:

Figura 3.1 – Diagrama de blocos do desenvolvimento do projeto



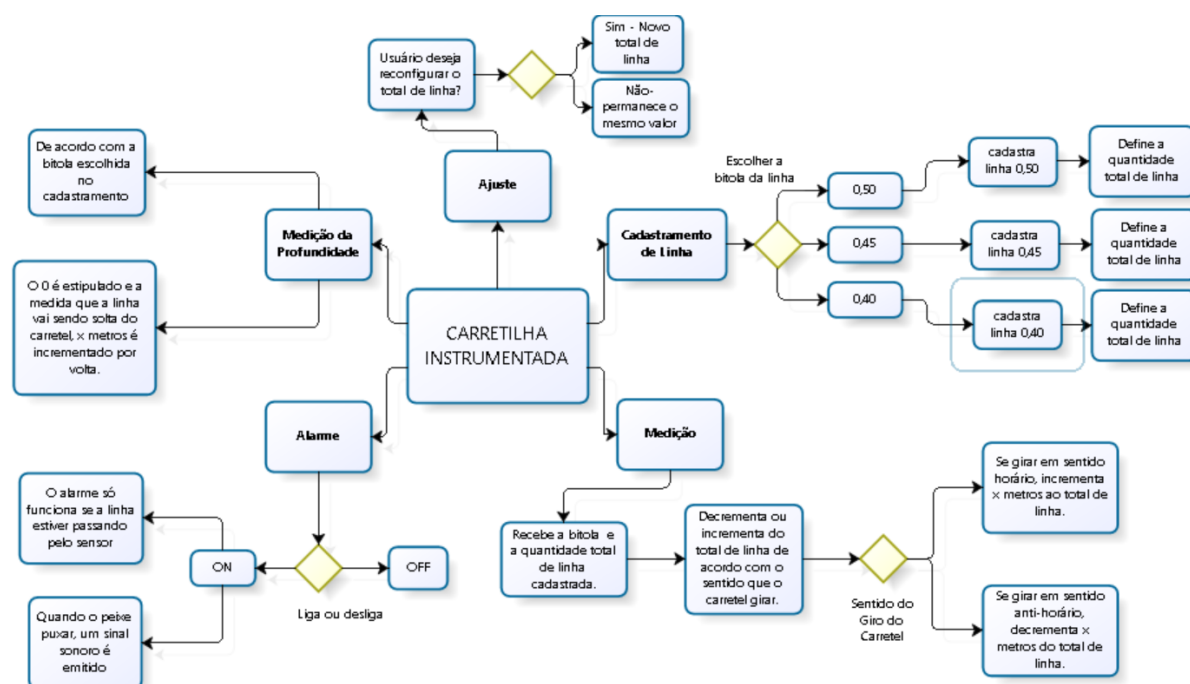
Fonte: do autor.

O desenvolvimento é iniciado na pesquisa e estudo dos componentes que envolvem o projeto, tanto dos componentes de hardware, quanto da parte de software. Após, se inicia o desenvolvimento do *Hardware* do projeto na placa de ensaio e adaptação no chassi da carretilha para disposição inicial dos sensores de efeito hall. O próximo passo o é desenvolvimento do *software* que irá contemplar as funcionalidades da carretilha. No estágio seguinte, o circuito montado na placa de ensaio será transferido para uma placa de circuito impresso. Após a confecção da placa de circuito impresso, será feito a adaptação final no chassi da carretilha e fixação do protótipo na vara. O último estágio é a realização de testes no Hardware e software.

3.1 Apresentação da Proposta

O projeto propõe o desenvolvimento de um protótipo de carretilha instrumentada focado na pesca esportiva de espera e pesada, que o usuário terá funcionalidades concentradas em uma único lugar de comando que auxiliarão na prática do esporte, são elas: cadastramento de três bitolas de linha que poderão ser usadas na medição – 0,40mm, 0,45mm e 0,50mm; medição, em metros, tanto da linha presente no carretel, quanto da linha descarregada, mostradas em um visor LCD; Alarme que é ativado quando algum peixe puxa a linha; e medição da profundidade do local pescado . Para chegar a solução foi desenvolvido um software em linguagem do IDE do arduino baseado nas linguagens C e C++, uma placa de circuito impresso com os componentes necessários para o funcionamento dos sensores que farão a medição, utilizada uma *Shield* LCD com botões para comandar toda as funcionalidades, como pode ser observado na Figura 3.2, sensores de efeito Hall para coletar os dados utilizados nas funcionalidades e adaptado ao projeto um alarme de pesca. A linha utilizada na carretilha se restringe às de nylon(poliamida) de monofilamento, com elasticidade que pode variar entre 15% e 30%. (CREMONESI, 2016) A figura 3.2 apresenta uma visão geral das funcionalidades do projeto:

Figura 3.2 – Visão geral das funcionalidades da carretilha instrumentada



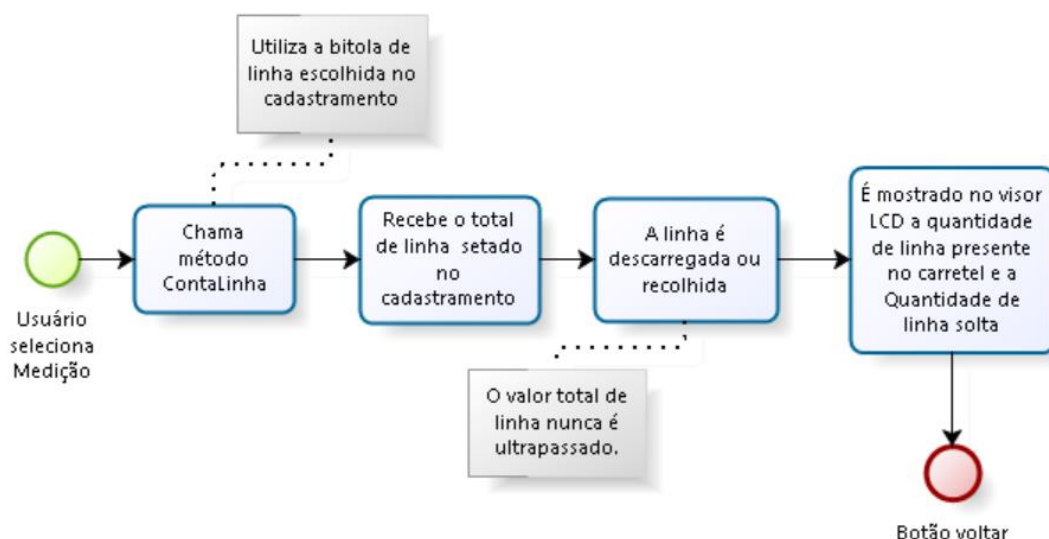
Fonte: do autor.

A lógica da quantidade de linha descarregada e a presente no carretel é feita por dois dos principais componentes do projeto: as funções `contalinhaCadastramento` e `contalinhaMedicao` presentes no programa da carretilha. Essas, são utilizadas no processo de cadastramento de nova linha, na medição e na medição da profundidade do local pescado. A medição que é feita por essas funções é realizada por dois sensores de efeito Hall que identificam quando um ímã, que está preso ao carretel, passa por eles. A cada vez que o ímã passar pelos sensores é contado uma volta, 360°. Essa solução possui dois sensores para que seja identificado qual o sentido que o carretel está girando, caso esteja girando no sentido horário, a cada vez que o ímã passar pelo sensor, a linha está sendo recolhida e assim é adicionado “x” metros ao total por volta, dependendo da bitola da linha. E caso o carretel esteja girando no sentido anti-horário, a linha está sendo descarregada e uma certa quantidade em metros é retirada do total.

Um desses processos que utiliza a função `ContaLinhaMedicao` é a medição, que é o utilizado para mostrar ao usuário no visor LCD a quantidade de linha que está presente no carretel e a quantidade que foi descarregada. O usuário por meio desse visor pode gerenciar esses valores durante a briga com o peixe, tendo a noção do quanto de linha falta puxar para

chegar ao peixe e quanto de linha ainda pode deixar ele descarregar antes dela acabar. A figura 3.3 apresenta esse processo:

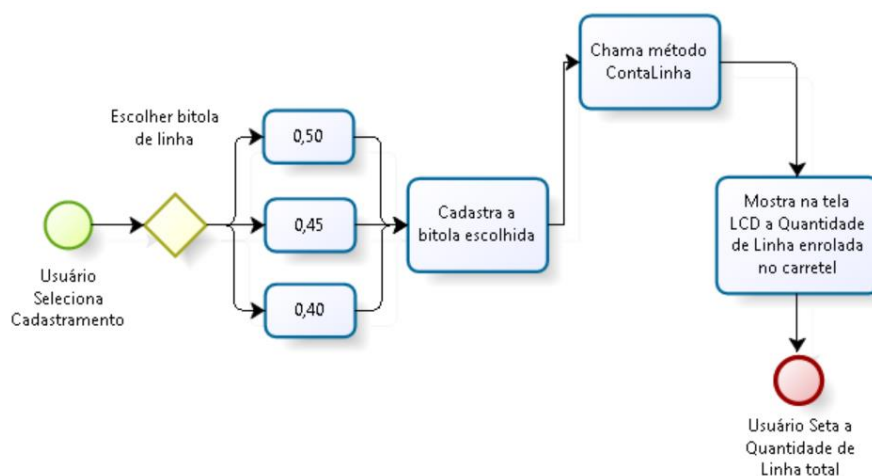
Figura 3.3 – Processo medição



Fonte: do autor.

No processo de cadastramento, o usuário escolhe a bitola de linha que ele deseja enrolar no carretel e a partir desse momento poderá começar a cadastrar a linha. Quando chegar a quantidade que ele deseja ter no carretel, esse valor deve ser setado, e assim, será salvo para que as outras funções o utilize. A medição precisa de três dados: a bitola de linha, pois com ela é que o programa saberá quantos metros de linha é puxada ou enrolada por volta; a quantidade total cadastrada, pois será dessa quantidade que se iniciará a pescaria; e a quantidade de voltas que o carretel deu, para saber quanto de linha será recolhida ou descarregada em determinada volta. A figura 3.4 apresenta esse processo:

Figura 3.4 – Processo cadastramento de linha

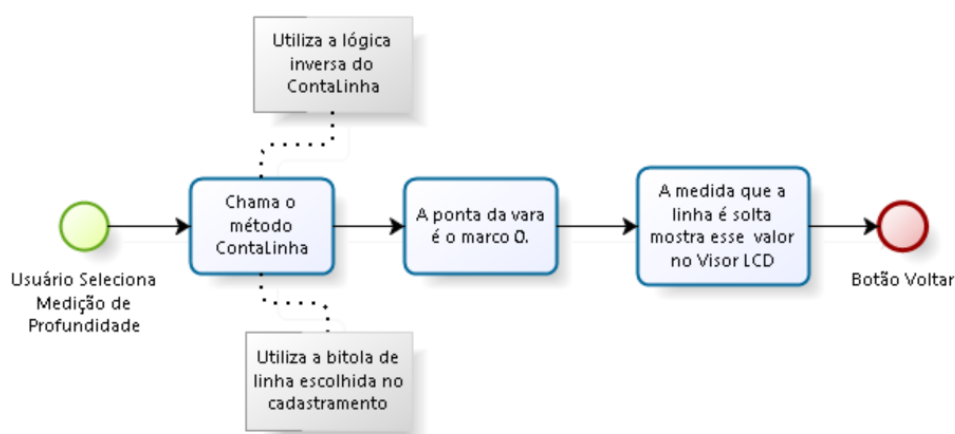


Fonte: do autor.

O processamento e armazenamento dos dados da quantidade de voltas lidas pelos sensores de efeito Hall são feitos pelo arduino Uno R3 e nele é acoplado uma *shield* LCD com botões, onde serão comandadas as funcionalidades da carretilha e mostradas suas informações.

O processo medição da profundidade depende da quantidade total de linha cadastrada no carretel e necessita da informação de qual bitola de linha que está sendo utilizada para que a função *ContaLinhaMedicao* possa retornar quanto de linha está sendo descarregada ou puxada por volta. A figura 3.5 ilustra o funcionamento desse processo:

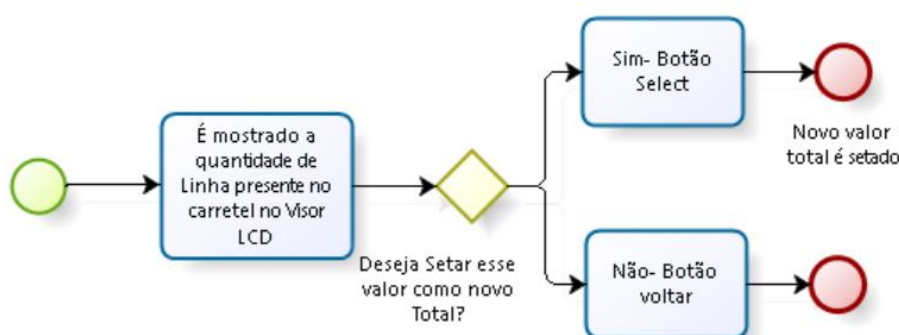
Figura 3.5 – Processo medição de profundidade



Fonte: do autor.

Suponhamos que foi cadastrado 100 metros de linha 0,40mm na carretilha, durante uma pescaria a linha se rompe e 10 metros são perdidos, sobrando 90 metros no carretel. Tendo em vista que cenários como esse podem ocorrer, para auxiliar o usuário a gerenciar a quantidade de linha total no carretel, a função ajuste mostra o quanto de linha que ainda está presente no carretel e questiona se o usuário deseja transformar esse valor no novo total de linha. A figura 3.6 mostra o funcionamento desse processo:

Figura 3.6 – Processo ajuste



Fonte: do autor.

A última funcionalidade que a carretilha instrumentada possui é o alarme. Pode ser desligado ou ligado por uma chave e o seu funcionamento depende que a linha esteja passando por ele. Estando essa devidamente posicionada, caso algum peixe a puxe, o sensor detecta essa tração e é emitido um sinal sonoro avisando o pescador de que algo está puxando.

Exceto o alarme, todas as outras funcionalidades podem ser acessadas por meio do *menu* da carretilha pela *shield* LCD com botões. Essa exceção teve de ser realizada pelo fato de que não ficaria funcional sair das outras funcionalidades para desligar ou ligar o alarme. O processo de funcionamento do Alarme é ilustrado na figura 3.7:

Figura 3.7 – Processo alarme

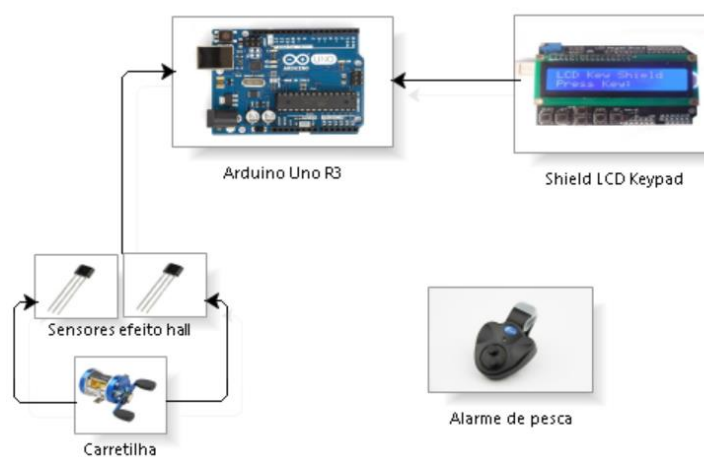


Fonte: do autor.

3.2 Componentes do Projeto

Neste subcapítulo explicaremos como cada componente se comporta dentro do sistema e qual sua função. A figura 3.8 mostra os componentes, em uma visão macro, que foram utilizados para desenvolver o projeto. Em seguida será aprofundado cada par de comunicação entre os componentes.

Figura 3.8 – Visão geral dos componentes do projeto



Fonte: do autor.

O projeto possui o alarme de pesca, que pode ser visualizado na figura 3.8, uma vez que a linha estiver posicionada, quando o peixe puxar utiliza um mecanismo que faz um *buzzer* emitir um sinal sonoro. Esse alarme foi inserido no projeto sem grandes alterações da forma que é encontrada no mercado.

A modificação foi realizada para que o pescador ligue e desligue o alarme por meio de um botão *ON/OFF* localizado na caixa de comando da carretilha.

3.2.1 Arduino Uno R3 e Shield LCD Keypad 1602

A *shield* LCD Keypad 1602 para arduino possui uma tela LCD de 16x2 com luz de fundo ajustável, botão de reset do arduino e cinco botões que foram programados para comandar as funcionalidades da carretilha. Nesse *shield* a conexão entre a tela LCD e o arduino é feita pelos pinos digitais 4,5,6,7,8 e 9, além do 10 que é utilizado para controle da luz de fundo. (ARDUINO E CIA, 2013) O quadro 3.1 apresenta as conexões entre o arduino e a *shield* e suas funções:

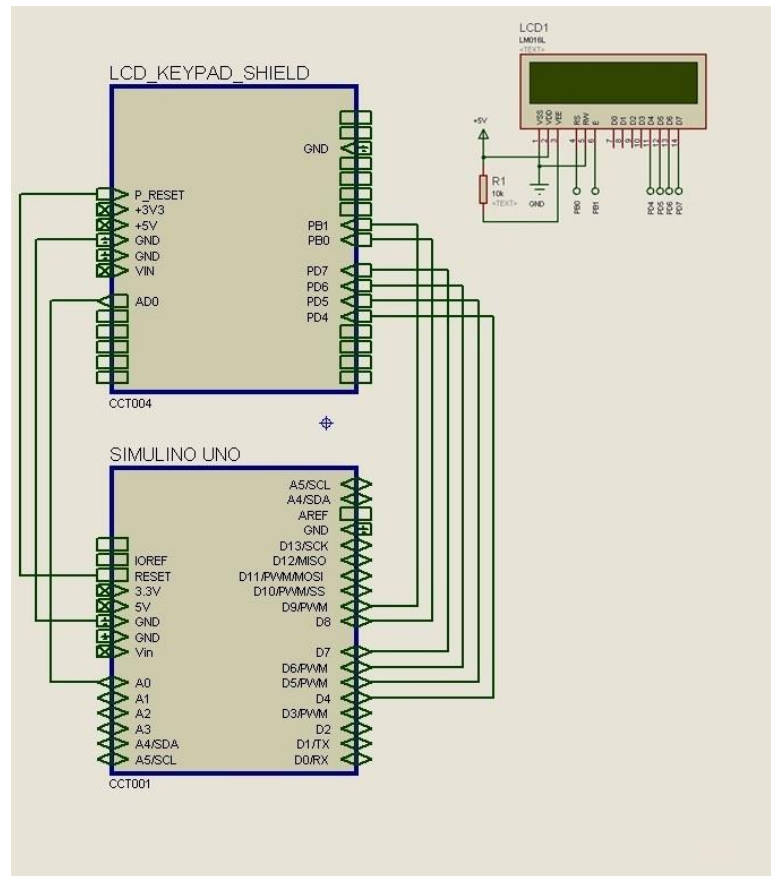
Quadro 3.1 - Pinos de conexão arduino e LCD keypad, e suas funções

Arduino-Pino	LCD Keydp	Função
4	DB4	DB4
5	DB5	DB5
6	DB6	DB6
7	DB7	DB7
8	RS	RS (dados ou seleção sinal display)
9	Enable	Enable
10	DB10	Controle da luz de fundo
A0	AD0	Botões(Select,up, right, down e left)

Fonte: adaptado de <http://arduinoormais.blogspot.com.br/2014/07/shield-lcd-keypad.html>

Para uma melhor visualização, a figura 3.9 representa uma simulação no software Proteus 8 o esquema da ligação entre o arduino Uno R3 e a *shield* LCD keypad:

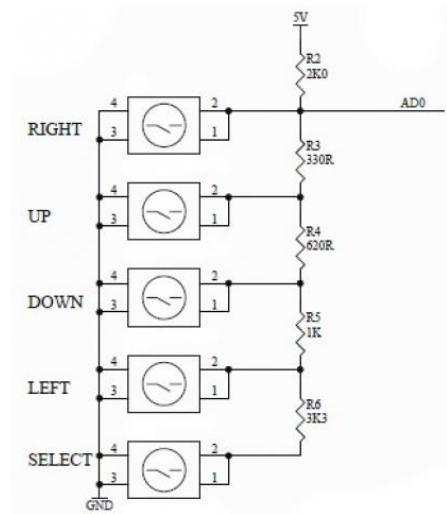
Figura 3.9 – Ligações shield LCD keypad e arduino UNO R3



Fonte: adaptado de <http://blogembarcado.blogspot.com.br/2013/06/simulino-v20-biblioteca-para-proteus.html>

Como a *shield* utiliza uma única porta analógica para os botões, para que se saiba qual tecla foi pressionada, ela utiliza uma associação de resistores que funciona como divisor de tensão. Quando é pressionada uma tecla, o arduino recebe o valor da tensão gerada e dentro do código há uma comparação para que se saiba qual tecla foi acionada. A figura 3.10 representa essa associação de resistores:

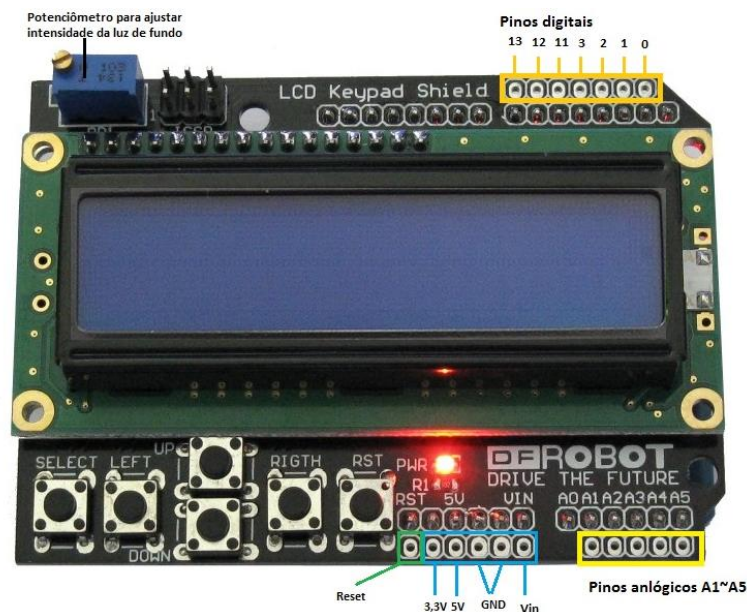
Figura 3.10 – Associação de resistores para acionar os botões



Fonte: <http://www.arduinoecia.com.br/2013/08/arduino-shield-lcd-16x2-com-keypad.html>

Como os pinos citados na figura 3.10 estão sendo utilizados para conexão entre os dois componentes, eles não podem ser utilizados como portas para outras aplicações, porém podemos utilizar as portas sobressalentes: as digitais 0,1,2,3,11,12 e 13; as analógicas A1, A2, A3, A4 e A5; e também, os pinos de energia – 5v, 3,3v e o terra (GND) (BACK, 2014). A figura 3.11 mostra esses pinos que podem ser utilizados:

Figura 3.11 – Pinos livres para uso shield LCD keypad 1602



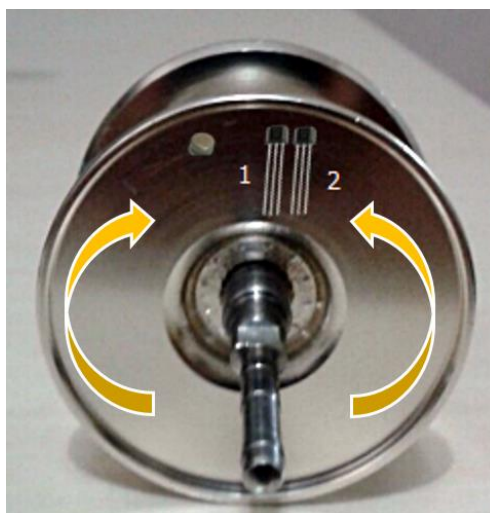
Fonte: do autor.

3.2.2 Carretilha e Sensor de Efeito Hall A3144

Os sensores de efeito Hall, modelo A3144, são responsáveis por identificar o campo magnético do ímã que está disposto no carretel da carretilha. Um super ímã de neodímio de 5 mm de diâmetro e 3 mm de altura foi colado de forma que quando o carretel girar ele passará cerca de 3mm de distância dos sensores. Essa distância foi encontrada levando os fatores em conta: a falta de espaço dentro da carretilha e o ponto em que os sensores de efeito Hall conseguem captar o campo magnético do ímã sem falhas de leitura.

Esses dois sensores foram dispostos juntos um ao lado do outro. Assim feito, para que saibamos qual o sentido que o carretel está girando. Exemplificando: caso o carretel esteja girando no sentido horário, o ímã precisará passar primeiro pelo sensor 1 e depois pelo sensor 2 para que o sistema reconheça que o giro é nesse sentido; e caso for identificado que o ímã passou primeiro pelo sensor 2 e depois pelo sensor 1, então o sistema sabe que o carretel está girando no sentido anti-horário. A figura 3.12 simula como acontecerá essa lógica.

Figura 3.12 – Carretel com ímã e sensores



Fonte: do autor.

A cada vez que o sistema obtém uma leitura de que o carretel está girando em um sentido, ou seja, quando o ímã passa pelos 2 sensores, é contado como uma volta.

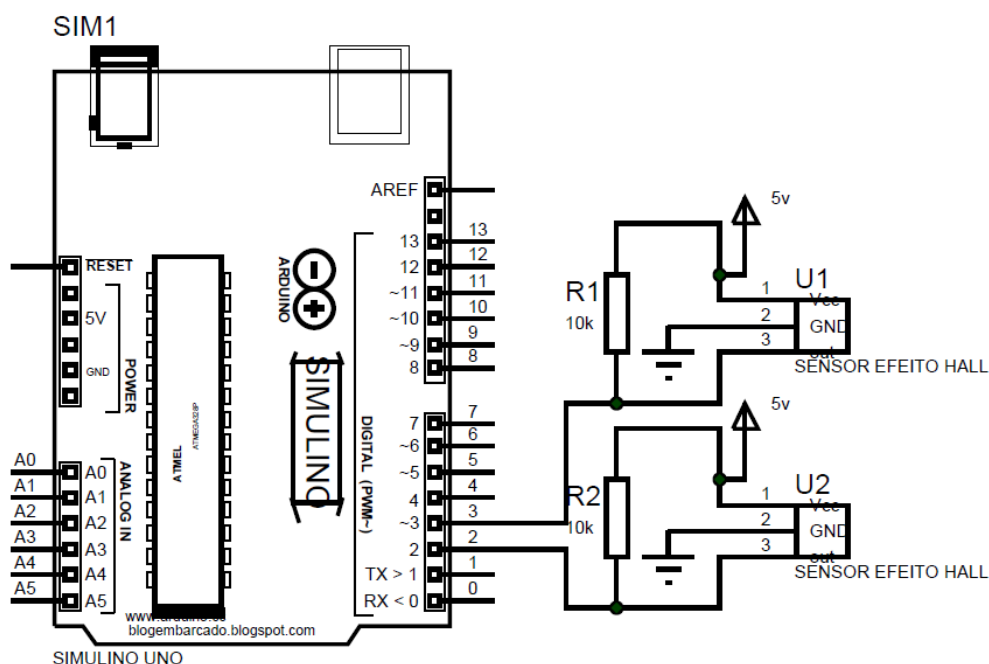
Nas funcionalidades da carretilha que utilizam a leitura do campo magnético do ímã pelos sensores de efeito Hall têm a seguinte lógica:

- Cadastramento e Medição: se o carretel estiver girando no sentido horário, um contador é incrementado a cada vez que o ímã passar pelos 2 sensores, e se estiver girando no sentido anti-horário é decrementado.
- Medição de profundidade: utiliza a lógica inversa supracitada, caso esteja girando no sentido anti-horário o contador é incrementado, e se girar no sentido horário é incrementado.

3.2.3 Sensor de Efeito Hall A3144 e Arduino Uno R3

Já foi explanado anteriormente, que o sensor de efeito Hall é responsável por detectar quando o ímã passa pelos sensores de forma que esse dado é enviado para o arduino Uno R3 e este faz o processamento, vinculando cada passagem pelos sensores como uma volta. Esses sensores, possuem 3 pernas de ligação, a primeira foi ligada na alimentação de 5v, a segunda no terra e a terceira em uma das portas sinal de saída digital. A figura 3.13 ilustra uma simulação, feita no software Proteus 8, da ligação entre os sensores e o arduino.

Figura 3.13 – Conexões sensores efeito hall e arduino UNO R3



Fonte: do autor.

Os sensores de efeito Hall enviam sinais digitais 1 (HIGH) ou 0 (LOW) para o arduino. Quando os sensores não detectam nenhum campo magnético suficiente para fazê-lo conduzir, o sinal enviado é 1 (HIGH) e quando o imã passa pelos sensores fazendo com que eles conduzam, o sinal enviado ao arduino é 0 (LOW).

3.3 Software do Projeto

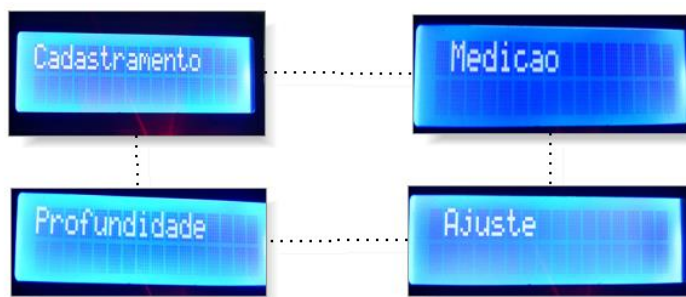
Será explicado nesse subcapítulo o funcionamento do programa utilizado na carretilha instrumentada, todo o código fonte está descrito no apêndice A.

O *software* do projeto foi programado no IDE do arduino, que é baseada nas linguagens de programação C e C++. Exceto o alarme de pesca, todas as outras funcionalidades foram implementadas por meio desse programa. Quando o usuário ligar o protótipo, por meio de uma chave ON/OFF será exibido na tela LCD um *menu* com as funcionalidades:

- Cadastramento de nova linha;
- Medição;
- Medição da Profundidade;
- Ajuste;

A figura 3.14 representa como será apresentado para o usuário as funções:

Figura 3.14 – Menu funções



Fonte: do autor.

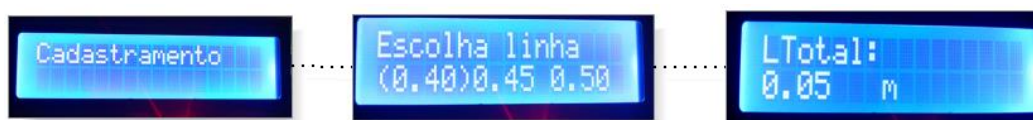
Os botões da *shield* LCD Keypad : *Up*, *Down*, *Right*, *Left* e *Select* ,serão usados para comandar as funcionalidades do projeto. Para que o usuário troque entre as opções do menu é

preciso apertar as teclas *Up* ou *Down* e se o *select* for pressionado a opção corrente é selecionada.

3.3.1 Cadastramento de Nova Linha

Quando o pescador for utilizar a carretilha instrumentada pela primeira vez e sempre que ele quiser cadastrar uma nova linha, a função que ele deverá entrar é no cadastramento de linha. Essa função é responsável por: mostrar por meio de um menu, quais bitolas de linhas podem ser cadastradas para que o usuário escolha; fazer a medição da quantidade que está sendo enrolada no carretel; e ao final o usuário definir a quantidade total de linha. A figura 3.15 representa os passos descritos acima na visão do usuário.

Figura 3.15 – Cadastramento

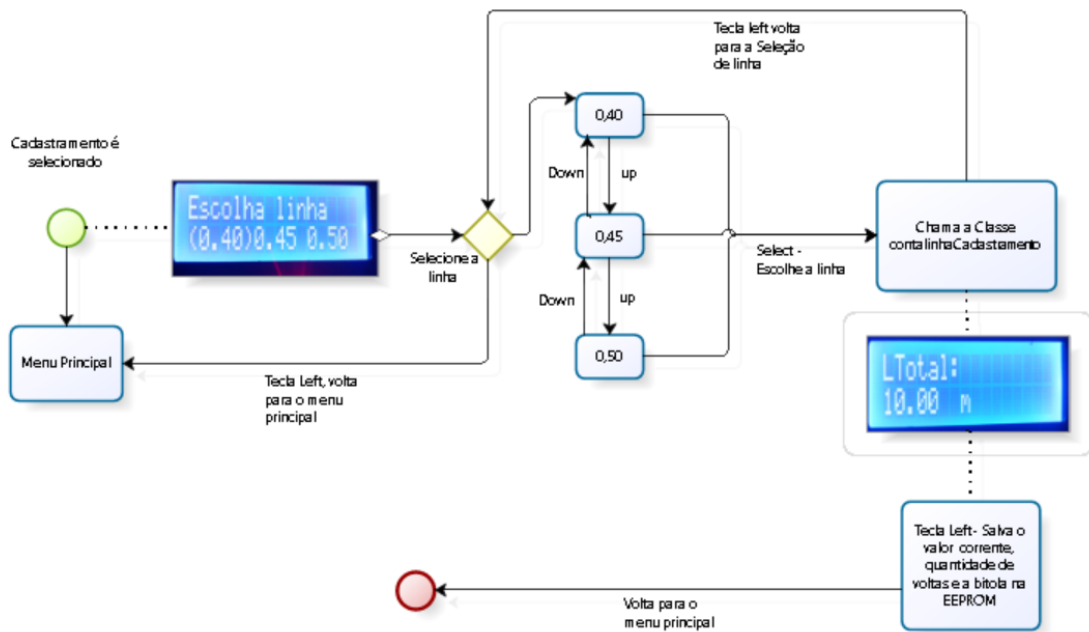


Fonte: do autor.

Como é possível observar na figura 3.15, depois que o cadastramento for selecionado, o próximo passo será determinar qual bitola de linha será enrolada no carretel, para isso, o usuário pode alternar entre as opções de bitola de linha com os botões *Up* e *Down*, caso esse primeiro seja pressionado os parênteses são movidos para a esquerda e o segundo para direita. Caso o botão *Select* seja pressionado, a bitola corrente é escolhida e a partir daí o usuário pode começar o cadastramento.

A medida que a linha vai sendo enrolada no carretel, o usuário poderá acompanhar o total que está sendo cadastrado como no exemplo da última tela da figura 3.15. No momento em que o usuário achar que a quantidade de linha cadastrada é o suficiente para sua pescaria, é preciso apertar a tecla *Left* para seja setado o total de linha, a quantidade de voltas e a bitola escolhida. Essa tecla também volta para o menu principal. A figura 3.16 representa o fluxograma do processo.

Figura 3.16 – Fluxograma cadastramento



Fonte: do autor.

Nesse processo, três informações são salvas na memória EEPROM do arduino quando o usuário apertar a tecla *Left*: a quantidade de linha total, a quantidade de voltas dada (contador) e a bitola escolhida. São salvas nessa memória pois ela é do tipo não volátil, assim, quando o sistema da carretilha instrumentada for desligado e ligado novamente, essas informações não serão perdidas. Essas informações são cruciais para as funções que fazem a medição de profundidade e a medição, pelo fato de as utilizarem como base para seus processos.

A medição da linha do cadastramento é feita pela função `contalinhaCadastramento`. Abaixo, é apresentado o código da função `contalinhaCadastramento` :

```

void contalinhaCadastramento(){

if (!digitalRead(pinosinal)){
    if (bloqs1 == 0){
        bloqs1 = 1;
        if (antihorario == 1){
            antihorario = 0;
            contador --;
            VoltasTabela = contador;
            MedicaoVolta();
            voltasC = voltasC - z;
            atdisp();
        }
        else
            horario = 1;
    }
}
else bloqs1 = 0;

if (!digitalRead(pinosinal2)){
    if (bloqs2 == 0){
        bloqs2 = 1;
        if (horario == 1){
            horario = 0;
            contador ++;
            VoltasTabela = contador;
            MedicaoVolta();
            voltasC = voltasC + z;
            atdisp();
        }
        else{
            antihorario = 1;
        }
    }
else bloqs2 = 0;
}

```

A função `contalinhaCadastramento` é chamada assim que o usuário escolhe a bitola de linha que deseja cadastrar, tem como propósito executar a lógica que foi descrita no item 3.2.2.

Quando os sensores de efeito Hall enviam 0 (LOW) para o arduino, refletindo que houve a detecção do campo magnético do imã, a comparação dos primeiros *if's* é válida e entra na sua estrutura. Caso a leitura tenha sido feita no sensor da direita, entra na estrutura do primeiro `if(!digitalRead(pinosinal))`, caso contrário, na estrutura do segundo `if(!digitalRead(pinosinal2))`.

As variáveis `bloq1` e `bloq2` funcionam como uma espécie de bloqueio para que o arduino não receba várias leituras do mesmo sensor. Então, por exemplo, se houver o imã ficar parado em cima do sensor1, ele fará apenas uma leitura do sensor 1 e espera até que haja a uma leitura do sensor 2.

Os if's que fazem a comparação dos sentidos, if (antihorario==1) e if (horario==1) seguem a seguinte lógica: caso haja a leitura primeiro do sensor1, então a variável horário é setada para 1 e espera até que haja a leitura do sensor2, havendo a leitura significa que o carretel está girando no sentido horário, o contador é incrementado e a variável horário retorna para 0. Se ocorrer o contrário, ou seja, a primeira leitura é do sensor2, a variável anti-horário é setada para 1 e quando houver a leitura do sensor1 significa que o sentido é o anti-horário, o contador é decrementado e a variável anti-horário retorna a 0.

Logo após o contador ser salvo na variável VoltaTabela e a função MedicaoVolta é chamada. Essa última possui uma tabela de if's que compara por meio da variável VoltaTabela em qual volta ela está localizada e retorna um valor z correspondente. O quadro 3.2 exemplifica um dos quadros de medição que foi inserida no programa:

Quadro 3.2 - Medição linha 0,50

Volta	Camada	Média das 3 medições
1~25	1	0,0463
26~47	2	0,0477
48~70	3	0,0484
71~96	4	0,0492
97~117	5	0,0503
118~144	6	0,0516
145~168	7	0,0525
169~195	8	0,0541
196~217	9	0,0555
218~241	10	0,0569
242~265	11	0,0581
266~288	12	0,0597
289~314	13	0,0613
315~338	14	0,0621
339~364	15	0,0637
365~388	16	0,0650
389~414	17	0,0665
415~438	18	0,0683
439~462	19	0,0696
463~484	20	0,0708

Volta	Camada	Média das 3 medições
485~510	21	0,0713
511~535	22	0,0724
536~558	23	0,0739
559~584	24	0,0757
585~608	25	0,0774
609~631	26	0,0781
632~657	27	0,0786
658~681	28	0,0793
682~705	29	0,0804
706~727	30	0,0814
728~754	31	0,0827
755~777	32	0,0841
778~804	33	0,0854
805~827	34	0,0871
828~854	35	0,0879
855~875	36	0,0887
876~900	37	0,0896
901~923	38	0,0909
924~947	39	0,0923
948~973	40	0,0932

Volta	Camada	Média das 3 medições
974~998	41	0,0944
999~1020	42	0,0952
1021~1046	43	0,0969
1047~1068	44	0,0986
1069~1094	45	0,1000
1095~1118	46	0,1011
1119~1142	47	0,1024
1143~1167	48	0,1037
1168~1192	49	0,1059
1193~1218	50	0,1078
1219~1243	51	0,1094
1244~1268	52	0,1110
1269~1292	53	0,1123
1293~1314	54	0,1134
>=1315	55	0,1144

Fonte: do autor

Esse valor z representa a terceira coluna da tabela. Com esse valor é realizado o cálculo $\text{voltasC} = \text{voltasC} + z$ para o sentido anti-horário ou $\text{volasC} = \text{voltasC} - z$ para o sentido horário, que representa a quantidade total de linha presente no carretel. A figura 3.17 ilustra como a função MedicaoVolta é implementada no programa:

Figura 3.17 – Implementação dos quadros no programa pela função *MedicaoVolta*

```

void MedicaoVolta( )
{
    if( y== 0.40)
    {
        if( VoltasTabela >= 1 & VoltasTabela <= 25)
            z= 0.0445;
        if( VoltasTabela >= 26 & VoltasTabela <= 52)
            z= 0.0460;
        if( VoltasTabela >= 53 & VoltasTabela <= 72)
            z= 0.0473;
        if( VoltasTabela >= 73 & VoltasTabela <= 97)
            z= 0.048;
        if( VoltasTabela >= 98 & VoltasTabela <= 120)
            z= 0.0482;
        if( VoltasTabela >= 121 & VoltasTabela <= 144)
            z= 0.0488;
        if( VoltasTabela >= 145 & VoltasTabela <= 170)
            z= 0.0494;
        if( VoltasTabela >= 171 & VoltasTabela <= 196)
            z= 0.0499;
        if( VoltasTabela >= 197 & VoltasTabela <= 218)
            z= 0.0515;
        if( VoltasTabela >= 219 & VoltasTabela <= 241)
            z= 0.0523;
        if( VoltasTabela >= 242 & VoltasTabela <= 266)
            z= 0.0531;
    }

    if( y== 0.45)
    {
        if( VoltasTabela >= 1 & VoltasTabela <= 28)
            z= 0.0446;
        if( VoltasTabela >= 29 & VoltasTabela <= 52)
            z= 0.0454;
        if( VoltasTabela >= 53 & VoltasTabela <= 77)
            z= 0.0463;
        if( VoltasTabela >= 78 & VoltasTabela <= 102)
            z= 0.0484;
        if( VoltasTabela >= 103 & VoltasTabela <= 126)
            z= 0.0493;
        if( VoltasTabela >= 127 & VoltasTabela <= 149)
            z= 0.0498;
        if( VoltasTabela >= 150 & VoltasTabela <= 172)
            z= 0.0512;
        if( VoltasTabela >= 173 & VoltasTabela <= 200)
            z= 0.0533;
        if( VoltasTabela >= 201 & VoltasTabela <= 222)
            z= 0.0558;
        if( VoltasTabela >= 223 & VoltasTabela <= 247)
            z= 0.0576;
        if( VoltasTabela >= 248 & VoltasTabela <= 270)
            z= 0.0584;
    }

    if( y== 0.50)
    {
        if( VoltasTabela >= 1 & VoltasTabela <= 25)
            z= 0.0463;
        if( VoltasTabela >= 26 & VoltasTabela <= 47)
            z= 0.0477;
        if( VoltasTabela >= 48 & VoltasTabela <= 70)
            z= 0.0484;
        if( VoltasTabela >= 71 & VoltasTabela <= 96)
            z= 0.0492;
        if( VoltasTabela >= 97 & VoltasTabela <= 117)
            z= 0.0503;
        if( VoltasTabela >= 118 & VoltasTabela <= 144)
            z= 0.0516;
        if( VoltasTabela >= 145 & VoltasTabela <= 168)
            z= 0.0525;
        if( VoltasTabela >= 169 & VoltasTabela <= 195)
            z= 0.0541;
        if( VoltasTabela >= 196 & VoltasTabela <= 217)
            z= 0.0555;
        if( VoltasTabela >= 218 & VoltasTabela <= 241)
            z= 0.0569;
        if( VoltasTabela >= 242 & VoltasTabela <= 265)
            z= 0.0581;
    }
}

```

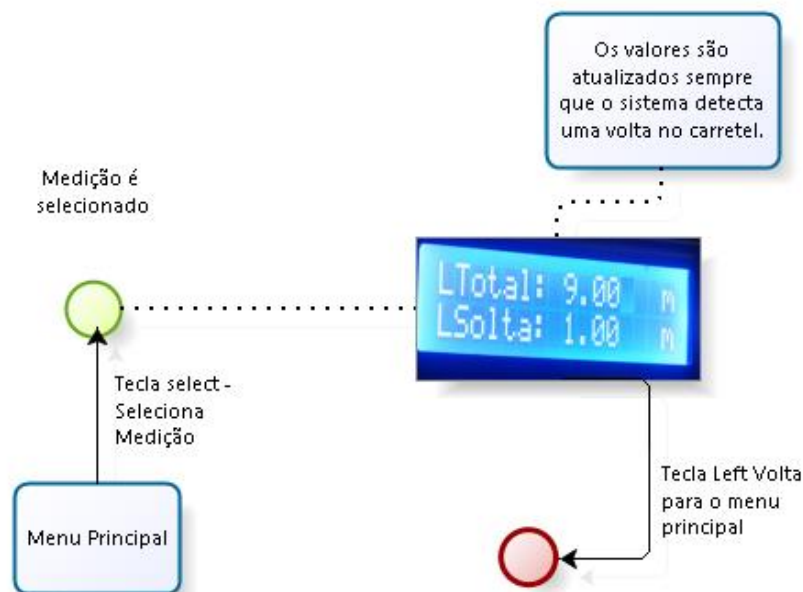
Fonte: do autor

Para atualização dos dados no display LCD, é utilizada uma função com um *switch case* para cada uma das funcionalidades que precisam que seus dados sejam atualizados a cada volta do carretel, sendo elas: cadastramento, medição e medição de profundidade.

3.3.2 Medição

A partir do momento que o sistema já detém as informações do total de linha e da bitola utilizada no cadastramento, o pescador já poderá começar a sua pescaria com a função medição. É mostrado no visor LCD a quantidade de linha que está presente no carretel e a quantidade de linha que foi solta. É setado no programa, para que o total de linha cadastrado nunca seja ultrapassado caso o pescador ainda continue girando a manivela da carretilha após ter recolhido toda a linha. A figura 3.18 representa o fluxograma desse processo medição:

Figura 3.18 – Fluxograma medição



Fonte: do autor.

É possível observar na figura 3.18 que no visor LCD são mostradas duas informações: Ltotal que representa a quantidade de linha no carretel e LSolta que representa a quantidade de linha que foi descarregada. A medida que a linha vai sendo descarregada ou enrolada os valores vão sendo atualizados por volta. LTotal é o valor total do cadastramento subtraído da quantidade de linha solta.

Essa função utiliza o método `contaLinhaMedicao` para saber quanto de linha por volta está sendo enrolada ou descarregada. O código da função `contalinhaMedicao` é apresentado:

```

void contalinhaMedicao(){
    if (!digitalRead(pinosinal)){
        if (bloqs1 == 0){
            bloqs1 = 1;

            if (antihorario == 1){
                antihorario = 0;
                contador2 --;
                VoltasTabela = contador + contador2;
                MedicaoVolta();
                voltas = voltas + z;
                atdisp();
            }
            else
                horario = 1;
        }
    }
    else bloqs1 = 0;

    if (!digitalRead(pinosinal2)){
        if (bloqs2 == 0){
            bloqs2 = 1;

            if (horario == 1){
                horario = 0;
                contador2 ++;
                VoltasTabela = contador + contador2;
                MedicaoVolta();
                voltas = voltas - z;

                atdisp();
            }
            else{
                antihorario = 1;
            }
        }
    }
    else bloqs2 = 0;
}

```

A lógica da função se assemelha bastante a da contalinhaCadastramento, a diferença é que a partir do momento que o usuário escolhe a funcionalidade medição e entra nessa função, a cada volta dada no carretel o contador que é incrementado ou decrementado é o contador2 e como a linha já foi cadastrada, a consulta a tabela de valores precisa ser feita do ponto em que o contador do cadastramento parou em diante, por esse motivo, o $\text{VoltasTabela} = \text{contador} - \text{contador2}$, retornando o z que refere-se a essa diferença.

A linha que é solta (LSolta) do carretel é mostrada pelo cálculo do $\text{voltas} = \text{voltas} + z$. Já a linha total é (LTotal) advém do cálculo $\text{totallinha} = \text{total} + \text{voltas}$, sendo o total, o valor total de linha salvo no cadastramento.

Quando o usuário perde uma quantidade de linha que se rompeu por algum motivo durante sua pescaria e entra no método ajuste para definir um novo total de linha, o que acontece

no programa é que é mostrado no visor LCD se o usuário deseja setar como novo total de linha o valor que está presente no carretel, que é igual ao totallinha(LTotal). A figura 3.19 ilustra a tela do visor LCD da função Ajuste:

Figura 3.19 – Funcionalidade ajuste visor LCD



Fonte: do autor

Caso o usuário aperte o botão *Right* é salvo na memória EEPROM esse novo valor e no programa é associado $\text{total} = \text{totallinha}$.

3.3.3 Medições da Profundidade

Nessa funcionalidade o usuário pode medir aproximadamente a profundidade do local pescado. A ponta da vara é colocada como referencial 0 e com uma chumbada (chumbo colocado na ponta da linha de pescar, antes do anzol) o pescador poderá descarregar a linha na vertical até que a chumbada chegue ao local que se deseja saber a profundidade. A cada volta no carretel é atualizado no visor LCD para que o usuário veja a quantidade de linha que está sendo descarregada. Utiliza a lógica inversa do contalinhaMedicao, ou seja, caso o carretel da carretilha esteja girando no sentido anti-horário, ao invés decrementar x metros por voltas, incrementa; e no sentido horário, decrementa.

O que é feito no programa para fazer a lógica inversa do contalinhaMedicao é inverter o valor de voltas. Sendo $\text{voltas} = -\text{voltas}$.

3.4 Medida do Valor das Voltas

Para implementar os valores, em metros, que cada volta correspondia no carretel, foi feito uma medição, para as 3 bitolas de linha, de cada camada que preenchida a medida que o carretel ia recebendo linha. Para isso, essa quantidade em metros foi mensurada em 3 pontos

do carretel, uma do lado direito, outra no meio e por fim do lado esquerdo. A figura 3.20 ilustra os locais onde as marcações foram realizadas.

Figura 3.20 – Marcações na linha do carretel



Fonte: do autor

Foi necessário fazer 3 medições por camada para que a média represente melhor como se comporta a variação do armazenamento de linha no carretel, pois a linha não é enrolada de forma uniforme pelo carretel. Como podemos observar na figura 3.20, o meio do carretel está mais cheio que as laterais, dessa forma, o valor medido nessa região é maior que nos cantos.

Após a linha ser marcada com uma caneta permanente, esse pedaço foi puxado para fora do carretel e medido com um paquímetro, retornando um valor em milímetros que é convertido em metros. A figura 3.21 representa esse processo:

Figura 3.21 – Medição da linha com paquímetro



Fonte: do autor

Depois de saber qual a medida média de cada camada, foi feito uma alteração no programa da carretilha para que mostrasse o valor do contador, que representa a quantidade de voltas dada no carretel, assim, esse dado foi utilizado para saber em cada camada a sua volta inicial e sua volta final.

Por fim, foram preenchidos quadros com esses valores. Quadros que se encontram no apêndice B.

3.5 Montagem do Protótipo

Para chegar a versão final do protótipo algumas etapas foram executadas, são elas:

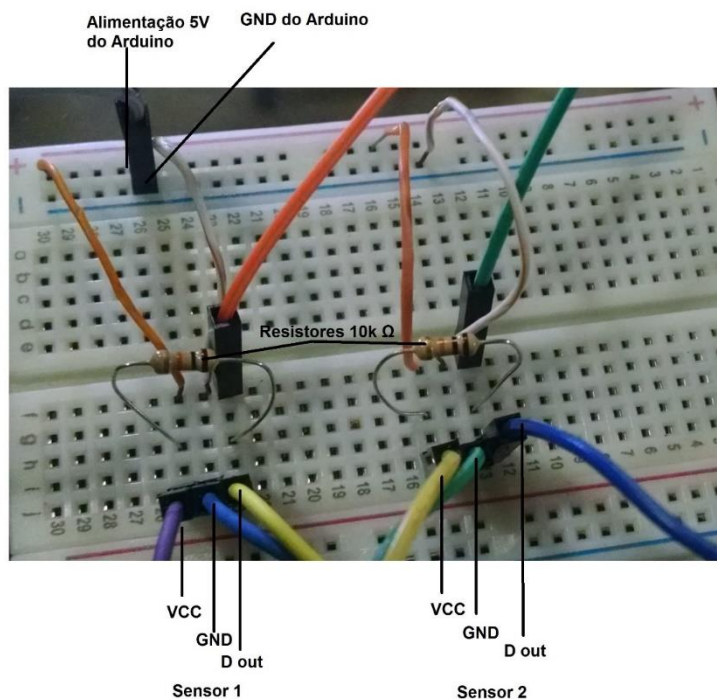
- Montagem do circuito utilizado na placa de ensaio (*proto board*) ;
- Adaptação no chassi da carretilha para dispor os sensores de efeito Hall;
- Transposição do circuito do projeto da placa de ensaio para uma placa de circuito impresso;
- Acomodação dos componentes em uma caixa e adaptação na vara de pesca para sua fixação e do alarme;

Cada processo descrito acima será aprofundado em subitens, de forma que o desenvolvimento e os fatores que interferiram neles sejam conhecidos.

3.5.1 Montagem do Circuito na Protoboard

Essa etapa consiste na montagem do circuito na placa de ensaio (*protoboard*), aqui os componentes ainda não são soldados, o que permitiu a realização de testes a fim de comprovar o correto funcionamento. A figura 3.22 representa o circuito do projeto na placa de ensaio.

Figura 3.22 – Circuito montado na protoboard



Fonte: do autor.

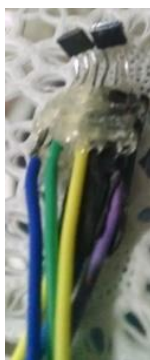
Esse circuito foi montado para o funcionamento dos dois sensores de efeito Hall. Podemos observar na figura 3.22 que a primeira perna de cada sensor está conectada a 5V do arduino, a segunda no terra (GND) e a terceira nos pinos digitais *out* do arduino, o primeiro sensor conectado no pino digital 2 e o segundo conectado no pino digital 3.

Os resistores de 10k estão em paralelo com as entradas de alimentação e digital de saída. Funciona como resistores de *Pull-up*, utilizado para evitar flutuação nos pinos de saída. A lógica é que quando um dos sensores detectar um campo magnético que o faça conduzir, o seu resistor limita a corrente que vai do VCC para o pino de saída e protege de eventuais tensões negativas.

3.5.2 Adaptação no Chassi da Carretilha

O chassi da carretilha teve de ser adaptado para dispor os sensores no correto lugar em que os mesmos conseguissem identificar o campo magnético do imã. Primeiramente, foram soldados nos fios e colados de forma que houvesse o mínimo de espaçamento entre eles, como pode-se observar figura 3.23.

Figura 3.23 – Sensores de efeito hall



Fonte: do autor.

O imã de neodímio de 5mm x 4mm foi preso ao carretel. Um suporte de alumínio foi colocado para estabilizar os sensores, depois que forem colados no chassi da carretilha, evitando que saíssem do lugar por algum fator externo. A figura 3.24 representa o resultado desse processo.

Figura 3.24 – Suporte para os sensores e disposição do ímã



Fonte: do autor.

Com o suporte e o ímã posicionado os sensores foram colados a uma distância que eles conseguissem detectar o campo magnético do ímã sem que afetasse no fechamento da tampa. A figura 3.25 ilustra o resultado desse processo.

Figura 3.25 – Adaptação no chassi da carretilha



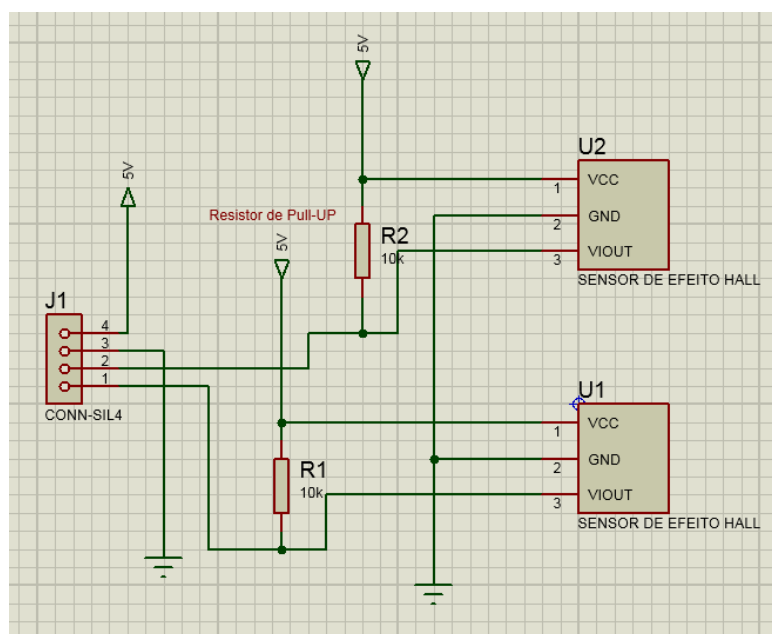
Fonte: do autor.

Foi cortado um pedaço da tampa para a passagem dos fios e em seguida todo o espaço foi preenchido com cola quente para fixar os componentes de maneira que qualquer movimento ou algo que esbarre nos fios não os tirem do lugar. Podemos observar a direita na figura 3.25 como os sensores ficaram dispostos internamente após a adaptação.

3.5.3 Placa de Circuito Impresso

Nessa etapa, o circuito montado na placa de ensaio foi transferido para uma placa de circuito impresso. A placa foi primeiramente projetada no ISIS do *software* Proteus 8. No ISIS os componentes que seriam utilizados foram dispostos, como é possível observar na figura 3.26.

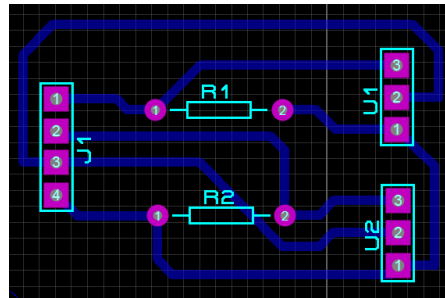
Figura 3.26 – Projeto placa de circuito impresso ISIS



Fonte: do autor.

Depois de montada no ISIS o próximo passo foi ajustar como os componentes ficariam configurados para impressão, criar e arrumar as trilhas de cobre de maneira que ficassem dispostas da melhor forma. Esse processo foi feito no *ARES* do *Proteus* e a figura 3.27 a placa já projetada.

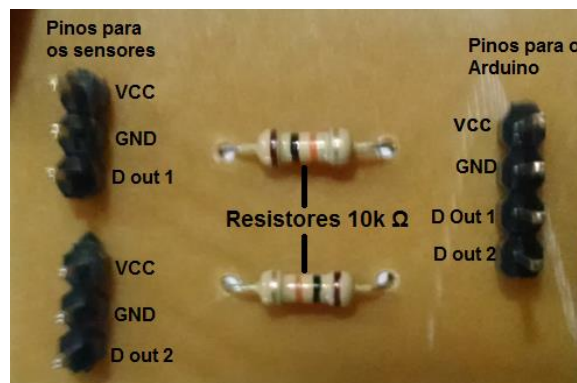
Figura 3.27 – Projeto placa de circuito impresso ARES



Fonte: do autor.

Dessa maneira, o circuito foi impresso numa placa de cobre que foi corroída restando apenas as trilhas. Os furos foram feitos em cima dos números que aparecem na figura acima para que os conectores e resistores fossem encaixados e soldados. A figura 3.28 representa o resultado desse processo.

Figura 3.28 – Placa de circuito impresso



Fonte: do autor.

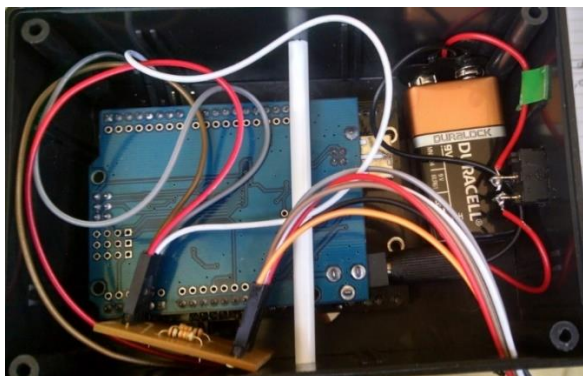
A dimensão final da placa é de 5 cm x 5 cm.

3.5.4 Montagem Final do Protótipo na Caixa e Fixação na Vara

Após todos os passos anteriores, o circuito impresso, a *shield* LCD keypad conectado ao arduino e uma bateria de 9V para alimentação do sistema foram encaixados dentro de uma caixa de plástico com dimensões 12 cm de comprimento 8 cm de largura e 5 cm de altura. Para

ligar o arduino foi inserido uma chave *ON/OFF* no fio positivo que vai da bateria para o arduino. A figura 3.29 representa como ficou dispostos os componentes dentro da caixa.

Figura 3.29 – Disposição dos componentes dentro da caixa de comando

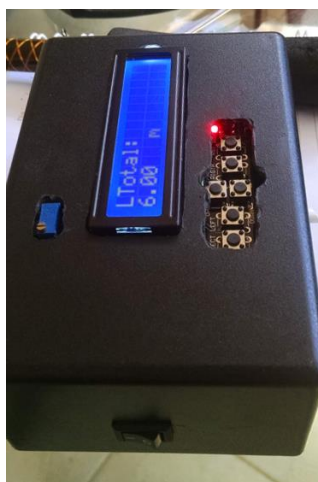


Fonte: do autor.

Como é possível observar na figura 3.29, um canudo passa por dentro da caixa, é por seu espaço interno que a linha vai correr. Essa adaptação foi necessária pois depois que a caixa foi fixada na vara, ela estava atrapalhando que a linha movimentasse no seu fluxo normal.

A caixa foi furada para que o *display* LCD ficasse a amostra juntamente com os botões de comando e seu potenciômetro. A figura 3.30 representa o resultado desse processo.

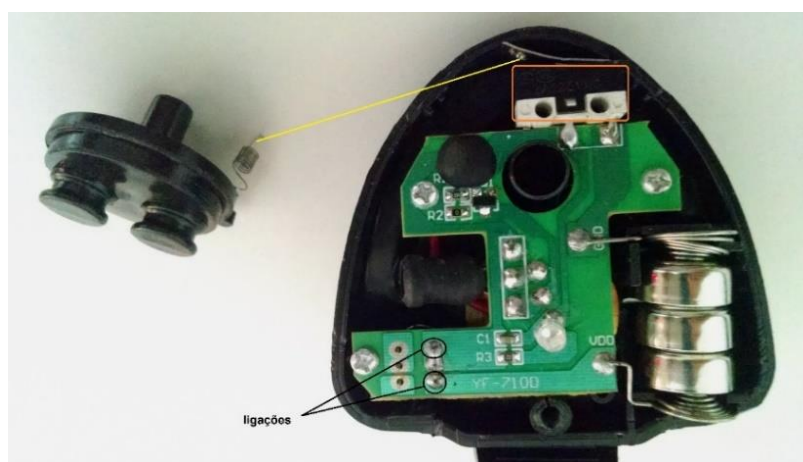
Figura 3.30 – Caixa de comando



Fonte: do autor.

Para ligar e desligar o alarme também foi adicionado uma chave *ON/OFF* na caixa de comando, a fim de que todas as funcionalidades da carretilha instrumentada fiquem concentradas em um local. A chave *ON/OFF* foi ligada na alimentação do alarme por dois fios substituindo o botão de pressão que nele já existia. A figura 3.31 representa a visão interna do alarme de pesca.

Figura 3.31 - Interior do alarme de pesca



Fonte: do autor.

Na parte inferior da figura 3.31 está representado onde foi ligado os fios da chave *ON/OFF*. Quando o alarme estiver ligado e um peixe puxar a linha, o mecanismo a esquerda que é responsável girar e acionar com a mola a micro chave KW10-B, destacada na figura na cor laranja. Apenas a primeira e a segunda perna da chave estão ligadas, indicando que está no estado normalmente aberto e só acionará o *led* e o *buzzer* caso a chave seja pressionada.

Por fim, a carretilha, a caixa de comando e o alarme foram fixados a vara. A figura 3.32 representa a montagem final do protótipo.

Figura 3.32 - Versão final do protótipo



Fonte: do autor.

4 TESTES E RESULTADOS

Este capítulo tem por objetivo apresentar os testes e resultados que foram executados no decorrer do desenvolvimento do projeto. Esses foram divididos em cenários para uma melhor organização e compreensão:

- a) **Primeiro cenário:** conferir se os sensores de efeito hall estão identificando as voltas realizadas pelo carretel.
- b) **Segundo cenário:** Comparação entre as medições realizadas pela carretilha instrumentada e a medição real na funcionalidade cadastramento.
- c) **Terceiro cenário:** Comparação entre as medições realizadas na carretilha para linha solta e a linha presente no carretel na funcionalidade medição com a medição real.

Ao final, após a apresentação dos cenários de teste e seus resultados, será mostrado o custo estimado de implementação do projeto.

4.1 Primeiro Cenário

4.1.1 Descrição

Esse cenário de teste tem por objetivo verificar se a dupla de sensores de efeito hall estão identificando o campo magnético do ímã sempre que esse passa por eles. Como a cada vez que o ímã passar pelos dois sensores é incrementado ou decrementado um contador, a depender do sentido do giro do carretel, foi feita uma simples modificação no código para que o valor dele fosse mostrado no visor LCD. A partir daí, podemos comparar o valor do contador com a ação de recolhimento que a carretilha *Caster 400* possui.

A carretilha *Caster 400* utilizada no projeto, possui uma ação de recolhimento de 5,3:1, ou seja, a cada volta dada na manivela o carretel gira aproximadamente 5,3 vezes. Como só é possível girar a manivela apenas no sentido horário, esse teste refere-se à quando a linha é recolhida e o contador incrementado.

4.1.2 Pré-requisitos

Alguns pré-requisitos são necessários para execução desse cenário de teste:

- Modificação no código para que o valor do contador fosse mostrado no visor LCD.
- A funcionalidade cadastramento deve ser escolhida, escolhendo qualquer uma das bitolas de linha, já que o contador é único. Deve ser escolhida essa funcionalidade pois foi nela onde foi feita a mudança no código para mostrar o valor do contador.

A figura 4.1 apresenta esse cenário de teste:

Figura 4.1 – Primeiro cenário de teste



Fonte: do autor.

4.1.3 Resultados

Como resultado desse teste foram obtidos os valores do contador da carretilha instrumentada, referente a quantidade de voltas dadas pelo carretel de acordo com cada volta completa na manivela.

Como a ação de recolhimento da carretilha é de 5,3 voltas no carretel para cada volta dada na manivela, os valores obtidos pelo contador foram comparados com os resultados do cálculo:

$$\text{Valor real} = \text{Voltas na manivela} \times 5,3 \quad (1)$$

A partir daí foi gerado o QUADRO 4.1:

Quadro 4.1 – Resultado primeiro cenário de teste

Identificação das voltas do carretel pelo protótipo		
Voltas Manivela	Voltas Reconhecidas	Voltas reais estimadas
1	5	5,3
2	10	10,6
3	16	15,9
4	21	21,2
5	26	26,5
6	31	31,8
7	36	37,1
8	42	42,4
9	47	47,7
10	52	53
20	105	106
30	157	159
40	210	212
50	262	265
60	315	318
70	367	371
80	420	424
90	472	477
100	525	530

Fonte: do autor.

Os resultados mostram que dentro de 100 voltas dadas na manivela da carretilha, todas as voltas foram identificadas pelos sensores de efeito hall, a diferença entre as voltas reconhecidas e as voltas reais estimadas se deram ao fato de que a ação de recolhimento especificada pelo fabricante é aproximada, não sendo exatamente 5,3 voltas no carretel a cada volta completa na manivela. Estima-se que o valor correto é de 5,25 voltas.

4.2 Segundo Cenário

4.2.1 Descrição

O segundo cenário consiste na aferição da quantidade de linha que é cadastrada e mostrada no visor LCD com a medição real. Para obter a medição real foi utilizado um medidor de linha de pesca (odômetro mecânico), sendo esse processo realizado em cada uma das três bitolas de linha.

A primeira medição é feita, aproximadamente, após 5 metros do início do cadastramento, a segunda após 10 metros e daí por diante de 10 em 10 metros.

4.2.2 Pré-requisitos

Para executar esse cenário de teste, alguns pré-requisitos são necessários:

- a) A funcionalidade cadastramento deve ser escolhida juntamente com a bitola de linha que se deseja cadastrar.
- b) A linha deve ser posicionada no odômetro mecânico de pesca.
- c) No processo de cadastrar a linha, essa deve ser pressionada para que seja enrolada com o mínimo de folga possível no carretel.

A figura 4.2 representa esse cenário de teste:

Figura 4.2 – Segundo cenário de teste



Fonte: do autor.

4.2.3 Resultados

Como resultado desse teste foi obtido os valores reais e comparados com valores mostrados no visor LCD e calculada porcentagem de erro relativo em cada momento de aferição. A % do erro relativo foi calculada pela seguinte formula:

$$Erro\ Percentual\ Relativo = \frac{(|Valor\ medido - Valor\ real|)}{Valor\ Real} \times 100\% \quad (2)$$

O valor real foi aferido por um odômetro mecânico que possui erro instrumental de $\pm 0,5$ mm, que é a metade do valor da menor medida, 1 cm. (MACEDO & Valerio, 2004)

O quadro 4.2 representa o erro da medição do cadastramento da bitola de linha 0,40mm:

Quadro 4.2 – Erro do cadastramento da linha 0,40

Precisão do Cadastramento		
Linha 0,40 mm		
Valor medido	Valor Real	% de Erro Relativo
5,05 m	5,10 m	0,98%
10,18 m	10,20 m	0,20%
20,10 m	20,20 m	0,50%
30,10 m	30,30 m	0,66%
40,05 m	40,40 m	0,87%
50,10 m	50,40 m	0,60%
60,05 m	60,30 m	0,41%
70,10 m	70, 20 m	0,14%
80,04 m	80,00 m	0,05%
90,03 m	89,90 m	0,14%
100,02 m	99,80 m	0,22%
110,03 m	109,70 m	0,30%
120,07 m	119,60 m	0,39%
130,02 m	129,50 m	0,40%
140,00 m	139,40 m	0,43%
150,03 m	149,30 m	0,49%
160,03 m	159,20 m	0,52%
170,02 m	169,10 m	0,54%
180,00 m	179,00 m	0,56%
184,70 m	183,60 m	0,60%
Total de linha cadastrada no teste = 184,70 m		

Fonte: do autor.

A quantidade de linha que é cadastrada na carretilha depende do gosto de cada pescador, para efeito de teste, foi cadastrado uma quantidade de linha que não deixasse o carretel tão cheio e nem tão vazio.

O quadro 4.3 representa a erro da medição do cadastramento da bitola de linha 0,45 mm:

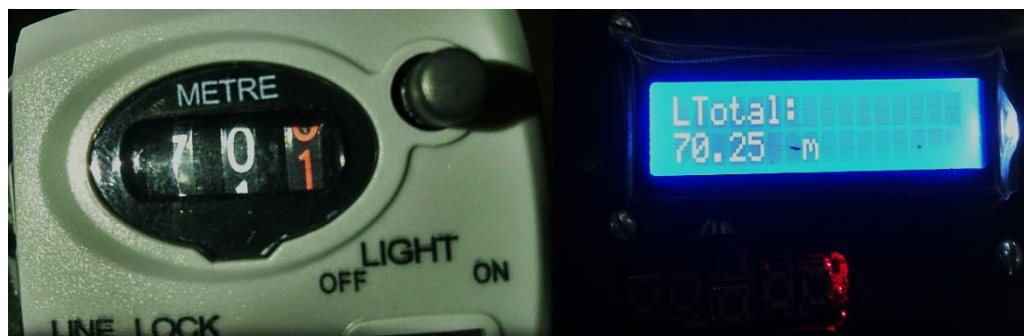
Quadro 4.3 – Erro do cadastramento da linha 0,45

Precisão do Cadastramento		
Linha 0,45 mm		
Valor medido	Valor Real	% de Erro Relativo
5,05 m	5,10 m	0,98%
10,03 m	10,10 m	0,69%
20,12 m	20,10 m	0,10%
30,17 m	30,20 m	0,10%
40,09 m	40,10 m	0,02%
50,06 m	50,10 m	0,08%
60,02 m	60,10 m	0,13%
70,10 m	69,90 m	0,29%
80,03 m	79,80 m	0,29%
90,00 m	89,70 m	0,33%
100,06 m	99,70 m	0,36%
110,00 m	109,60 m	0,36%
120,05 m	119,50 m	0,46%
130,02 m	129,50 m	0,40%
138,00 m	137,40 m	0,44%
Total de linha cadastrada no teste = 138,00 m		

Fonte: do autor

Os valores referentes a aferição por volta dos 70 metros da carretilha e comparados com valor real estão representados na figura 4.3 como um exemplo de como os testes foram realizados.

Figura 4.3 – Valores aferidos 70m na carretilha e odômetro mecânico linha 0,45



Fonte: do autor

Por fim, o quadro 4.4 representa o último teste realizado nesse cenário:

Quadro 4.4 – Erro do cadastramento da linha 0,50

Precisão do Cadastramento		
Linha 0,50 mm		
Valor medido	Valor Real	% de Erro Relativo
5,00 m	4,90 m	2,04%
10,00 m	9,80 m	2,04%
20,06 m	19,80 m	1,31%
30,02 m	29,70 m	1,08%
40,03 m	39,60 m	1,09%
50,10 m	49,50 m	1,21%
60,00 m	59,50 m	0,84%
70,06 m	69,60 m	0,66%
80,01 m	79,60 m	0,52%
90,23 m	89,80 m	0,48%
100,00 m	99,50 m	0,50%
102,15 m	101,60 m	0,54%
Total de linha cadastrada no teste = 102,15 m		

Fonte: do autor

Alguns fatores que interferem na precisão das medições foram identificados:

- A cada novo cadastramento a linha é enrolada de uma maneira, algumas vezes com menos folga no carretel, outras com a linha mais frouxa. A depender de como a linha cadastrada, aumenta ou diminui o valor em metros de cada volta. Essa variação pode causar um erro e é indicado para o diminuir que esse processo seja feito com a menor quantidade de folga possível, da mesma maneira feita no processo de medição.
- Erro instrumental dos utensílios utilizados na medição.
- Apesar da carretilha *Caster 400* possuir um guia de linha, a linha não é acomodada de forma uniforme no carretel. Nos testes de cadastramento, por vezes o meio do carretel possuía um maior volume comparado aos cantos.

4.3 Terceiro Cenário

4.3.1 Descrição

Neste cenário de teste, é utilizado os valores totais de linha cadastrada do cenário de teste anterior e a partir desses valores é aferido a quantidade de linha solta do carretel e calculado a porcentagem de erro relativo. A medida que a linha é solta é possível calcular a quantidade de linha presente no carretel.

A primeira medição é feita, aproximadamente, após 5 metros do início da funcionalidade medição, a segunda após 10 metros e daí por diante de 10 em 10 metros.

4.3.2 Pré-requisitos

Para execução desse cenário de teste alguns pré-requisitos são necessários:

- O total de linha já deve ter sido cadastrada no cenário de teste anterior.
- A linha deve ser posicionada no odômetro mecânico de pesca.
- Deve-se entrar na funcionalidade medição.

A figura 4.1 do cenário teste anterior também representa este.

4.3.3 Resultados

Os valores reais obtidos no teste foram comparados com os valores calculados pela carretilha instrumentada. Os resultados desse teste foram representados em quadros.

A porcentagem de erro relativo foi calculada com a mesma formula que a do cenário anterior. Aqui, foram calculados a porcentagem de erro da linha solta e a porcentagem de erro da linha presente no carretel. O quadro 4.5 representa o resultado dos testes na linha de bitola 0,40 mm:

Quadro 4.5 – Erro da funcionalidade medição linha 0,40 mm

Erro da linha Solta e Presente no carretel da Funcionalidade Medição					
Linha 0,40 mm					
Valor medido Lsolta	Valor Real	% de Erro Relativo	Valor Ltotal calculado	Valor Real calculado	% Erro Relativo
5,30 m	5,15 m	2,91%	179,40 m	177,85 m	0,87%
11,00 m	10,80 m	1,85%	173,70 m	172,20 m	0,87%
20,30 m	19,70 m	2,01%	164,40 m	163,10 m	0,80%
30,15 m	29,70 m	1,52%	154,55 m	153,30 m	0,82%
40,12 m	39,60 m	1,31%	144,58 m	143,40 m	0,82%
50,30 m	49,50 m	1,62%	134,40 m	133,50 m	0,67%
60,05 m	59,50 m	0,92%	124,65 m	123,50 m	0,93%
70,06 m	69,40 m	0,95%	114,64 m	113,60 m	0,92%
80,00 m	73,30 m	0,88%	104,70 m	103,70 m	0,96%
90,02 m	89,2 m	0,92%	94,68 m	93,80 m	0,94%
100,03 m	99,10 m	0,94%	84,67 m	83,90 m	0,92%
110,00 m	109,00 m	0,92%	74,70 m	74,00 m	0,95%
120,20 m	118,80 m	1,18%	64,50 m	64,20 m	0,47%
130,05 m	128,70 m	1,05%	54,65 m	54,30 m	0,64%
140,05 m	137,70 m	0,97%	44,65 m	44,30 m	0,79%
150,00 m	148,60 m	0,94%	34,70 m	34,40 m	0,87%
160,01 m	158,60 m	0,89%	24,69 m	24,40 m	1,19%
170,08 m	168,50 m	0,94%	14,62 m	14,50 m	0,83%
180,01 m	178,5 m	0,90%	4,60 m	4,50 m	2,22%
184,70 m	183	0,93%	0,00 m	0,00 m	0,00%

Fonte: do autor.

Como podemos observar no quadro acima, o maior erro relativo percentual quando a linha está sendo solta do carretel foi de 2,91% e que em foram medidos com toda a linha solta 183 metros pelo odômetro mecânico, enquanto a carretilha identificou 184,70 metros. Isso gera um erro de 0,93% o que não afeta no desempenho da pescaria.

A linha presente no carretel se comportou de forma parecida, excetuando o erro percentual relativo de 2,22%, o restante das medições ficaram em torno de 1% de erro.

O valor de L_{total} é calculado pela seguinte formula:

$$L_{total} = Valor\ Total\ Cadastrado - Valor\ Lsolta \quad (3)$$

O valor total de linha cadastrado é referente a quantidade de linha enrolada no cenário de teste anterior. O valor real de linha presente no carretel foi calculado pela formula:

$$Valor\ Real = Valor\ Real\ total\ medido - Valor\ Real\ medido\ de\ Lsolta \quad (4)$$

Após o teste executado na bitola de linha 0,40, foi realizado o mesmo processo na bitola de linha 0,45. O quadro 4.6 representa os resultados:

Quadro 4.6 – Erro da funcionalidade medição linha 0,45 mm

Erro da linha Solta e Presente no carretel da Funcionalidade Medição					
Linha 0,45 mm					
Valor medido	Valor Real	% de Erro Relativo	Valor L_{total} calculado	Valor Real calculado	% Erro Relativo
5,30 m	5,20 m	1,92%	132,70 m	131,70 m	0,76%
10,46 m	10,30 m	1,55%	127,54 m	126,60 m	0,74%
20,04 m	19,90 m	0,70%	117,96 m	117,00 m	0,82%
30,20 m	30,00 m	0,66%	107,30 m	106,00 m	0,85%
40,04 m	39,60 m	1,11%	97,96 m	97,30 m	0,68%
50,00 m	49,50 m	1,01%	88,00 m	87,40 m	0,69%
60,00 m	59,50 m	0,84%	78,00 m	77,40 m	0,78%
70,00 m	69,40 m	0,86%	68,00 m	67,50 m	0,74%
80,00 m	79,40 m	0,76%	58,00 m	57,50 m	0,87%
90,00 m	89,30 m	0,78%	48,00 m	47,60 m	0,84%
100,12 m	99,70 m	0,93%	37,88 m	37,70 m	0,48%
110,00 m	109,10 m	0,82%	28,00 m	27,80 m	0,72%
120,05 m	119,00 m	0,88%	17,95 m	17,90 m	0,28%
130,07 m	129,00 m	0,83%	7,93 m	7,90 m	0,38%
138,00 m	136,90 m	0,80%	0,00 m	0,00 m	0,00%

Fonte: do autor.

Como podemos observar no quadro acima, o maior erro relativo percentual quando a linha está sendo solta do carretel foi de 1,92% e que em foram medidos com toda a linha solta 136,90 metros pelo odômetro mecânico, enquanto a carretilha identificou 138,00 metros. Assim como no teste da linha 0,40, esse pequeno erro de 0,80% não afeta no desempenho da pescaria.

De forma parecida, também, a linha presente no carretel tem um erro percentual relativo máximo de 2,22% e o restante das medições ficaram com um erro em torno de 0,70% de erro.

Por fim, foi realizado testes na bitola de linha 0,50. Os resultados estão representados no quadro 4.7:

Quadro 4.7 – Erro da funcionalidade medição linha 0,50 mm

Erro da linha Solta e Presente no carretel da Funcionalidade Medição					
Linha 0,50 mm					
Valor medido LSolta	Valor Real	% de Erro Relativo	Valor Ltotal calculado	Valor Real calculado	% Erro Relativo
5,38 m	5,30 m	1,51%	96,77 m	95,9 m	0,91%
10,00 m	9,80 m	2,04%	92,15 m	91,4 m	0,82%
20,00 m	19,70 m	1,52%	82,15 m	81,5 m	0,80%
30,00 m	29,70 m	1,01%	72,15 m	71,5 m	0,91%
40,03 m	39,70 m	0,83%	62,12 m	61,5 m	1,01%
50,04 m	49,90 m	0,80%	51,85 m	51,3 m	0,68%
60,20 m	59,80 m	0,67%	41,95 m	41,4 m	1,33%
70,10 m	69,60 m	0,72%	32,05 m	31,6 m	1,42%
80,10 m	79,50 m	0,75%	22,05 m	21,7 m	1,61%
90,00 m	90,00 m	1,00%	11,25 m	11,2 m	0,45%
102,15 m	101,2 m	0,94%	0 m	0 m	0,00%

Fonte: do autor.

Os testes de linha solta da linha 0,50 mm mostra um resultado de erros que estão em torno de 1%, com erro máximo de 2,04%. Como é possível observar no quadro acima, quando toda a linha é solta, é medido o valor real 101,20 m enquanto a carretilha apresentou 102,15 m com um erro de 0,94%.

O valor Ltotal calculado teve um erro percentual que não ultrapassou 1,61%.

Nesse cenário de teste, também é identificado fatores que interferem na precisão das medições foram identificados, alguns similares ao cenário anterior:

- Além dos fatores que interferiram na precisão das medições, esse cenário apresentou mais um fator. Como a funcionalidade medição, após a linha cadastrada, é aquela em que o usuário utilizará enquanto estiver pescando, à medida que a linha vai sendo solta e recolhida o erro referente a folga ao enrolar a linha no carretel pode aumentar ou diminuir.
- Erro instrumental dos utensílios utilizados na medição.

- Apesar da carretilha *Caster 400* possuir um guia de linha, a linha não é acomodada de forma uniforme no carretel. Por esse motivo, na medição, descrita no subcapítulo 3.4 do desenvolvimento, para saber a quantidade de linha que é enrolada ou descarregada por volta, foi feita uma média de 3 medições, uma em cada canto e outra no meio.

4.4 Custo Estimado do Projeto

O quadro 4.8 apresenta o custo estimado dos componentes utilizados para a implementação do projeto, considerando preços de mercado.

Quadro 4.8 – Custo do projeto

Custo de implementação do projeto		
Componentes do projeto		
Descrição	Quantidade	Preço
Placa Arduino UNO R3	1	R\$ 37,89
Shield LCD Keypad 1602	1	R\$ 23,99
Carretilha Caster 400	1	R\$ 130,00
Alarme de Pesca	1	R\$ 22,99
Caixa Protetora de Plástico	1	R\$ 20,00
Sensor de Efeito Hall	2	R\$ 7,80
Chave ON/OFF	2	R\$ 3,00
Jumper 20 cm	20	R\$ 11,00
Resistor 10k Ohms 1/4W	2	R\$ 0,25
Protoboard 400 furos	1	R\$ 14,90
Imã de Neodímio 5mm x 3mm	1	R\$ 0,85
Bateria 9v	1	R\$ 13,90
Vara de Pesca 1,68 m 15-30 lb	1	R\$ 65,00
Linha 0,40 mm Dourado 100m	2	R\$ 8,00
Linha 0,45 mm Dourado 100m	2	R\$ 10,00
Linha 0,50 mm Dourado 100m	1	R\$ 8,00
Suporte Para Vara	3	R\$ 13,00
Parafuso	4	R\$ 0,50
	TOTAL	R\$ 391,07

Fonte: do autor.

5 CONSIDERAÇÕES FINAIS

5.1 Conclusão

Com o avanço da tecnologia, até mesmo atividades bastante antigas como a pesca vem se desenvolvendo, tanto em referência a seus equipamentos, quanto no modo de pescar com uma maior consciência ambiental. A proposta desse trabalho tem relação com o desenvolvimento de equipamentos para a pesca com objetivo de aumentar a eficácia na captura dos peixes.

Com foco na pescaria de espera e pesada, funcionalidades foram implementadas de modo que o usuário tenha informações que o ajudarão na pescaria com uma maior comodidade sem perder a esportividade. Para implementar essas funcionalidades foi necessário a construção de um protótipo que envolvia hardware e software.

A metodologia do desenvolvimento utilizada permitiu atingir aos objetivos geral e específicos. A construção inicial do hardware na *protoboard* possibilitou que os componentes utilizados fossem testados identificando a necessidade de ajustes no sistema. A utilização do arduino Uno R3 como microcontrolador é adequada, já que a quantidade de exemplos de códigos online e na sua própria IDE ajudou a amenizar a complexidade do desenvolvimento do código da carretilha instrumentada.

Os resultados dos testes foram satisfatórios, pois o erro percentual relativo das medições de linha teve máximo de 2,91%, o que não afeta no desempenho da pescaria.

Os sensores de efeito hall se mostraram alinhados a proposta do trabalho, uma vez que sempre que detectavam o campo magnético do ímã enviava para o arduino essa informação.

O trabalho proposto se mostrou complexo em algumas áreas, destacando o software e a parte mecânica de adaptação do chassi da carretilha. Apesar disso, o protótipo proposto se mostrou em condições adequadas e eficazes para a pesca.

5.2 Propostas de Trabalhos Futuros

No decorrer do desenvolvimento do trabalho proposto, alguns pontos foram identificados de melhoria para trabalhos futuros.

A primeira proposta é ter uma maior preocupação com a impermeabilidade do projeto, uma vez que, apesar de não ocorrer com muita frequência, o equipamento pode cair dentro d'água e os componentes internos serem comprometidos.

Outra proposta seria a confecção de um suporte para a caixa de comando, de modo que essa pudesse ser encaixada em qualquer vara de pesca compatível com a modalidade de pesca pesada de espera.

Por último, a implementação uma formula matemática que representasse a quantidade de linha que está sendo enrolada ou descarregada, com objetivo de diminuir o erro.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 Adolpho, G. (24 de Agosto de 2011). *Sensor de efeito hall*. Acesso em 15 de Abril de 2016, disponível em eletronicaegames: <http://eletronicaegames.blogspot.com.br/2011/08/sensores-de-efeito-hall.html>

- 2 Allego MicroSystem. (2016). *3144 SENSITIVE HALL-EFFECT SWITCHES*. Acesso em 18 de Abril de 2016, disponível em alldatasheet: <http://pdf1.alldatasheet.com/datasheet-pdf/view/55092/ALLEGRO/A3144.html>

- 3 Arduino. (2015). *Arduino.cc*. Acesso em 04 de Abril de 2016, disponível em ARDUINO: <https://www.arduino.cc/en/Guide/Introduction>

- 4 Arduino e Cia. (27 de Agosto de 2013). *arduino-shield-lcd-16x2-com-keypad*. Acesso em 17 de 05 de 2016, disponível em Arduino e Cia: <http://www.arduinoecia.com.br/2013/08/arduino-shield-lcd-16x2-com-keypad.html>

- 5 Back, M. (23 de Julho de 2014). *Shield LCD Keypad*. Acesso em Abril 13 de 2016, disponível em Arduino Mais: <http://arduinoMais.blogspot.com.br/2014/07/shield-lcd-keypad.html>

- 6 Bensusan, D. J. (2008). *Bíblia do Pescador*. São paulo: Zillig.

- 7 Cremonesi. (08 de 07 de 2015). *Conheça os tipos de carretilha disponíveis no mercado*. Acesso em 04 de 03 de 2016, disponível em Cremonesi: <http://blog.cremonesi.com.br/conheca-os-tipos-de-carretilha-disponiveis-no-mercado/>

- 8 David, M. (5 de 11 de 2009). *Pesca e Turismo*. Acesso em 03 de 04 de 2016, disponível em Fishingtur: <http://www.pescaeturismo.com.br/pescaria/carretilhas-conheca-e-arremesse-sem-cabeleiras/>

- 9 Fabichak, I. (1995). *Pantanal: a pesca esportiva*. São Paulo: Livraria Nobel S.A.

- 10 Fukumoto, S. (1994). *Noções Gerais Da Pesca De Arremesso*. São Paulo: Zillig.

- 11 Garcia, J. H. (2013). *infoEscola*. Acesso em 02 de 04 de 2016, disponível em InfoEscola navegando e aprendendo: <http://www.infoescola.com/economia/atividade-pesqueira/>

- 12 *HackLab Belenzinho*. (2015). Acesso em 04 de Abril de 2016, disponível em HackLeste: <https://hackleste.wordpress.com/arduino/>

- 13 IBAMA. (2008). *Guia de Pesca Amadora*. Brasília.

- 14 Lopes, M. (17 de Novembro de 2013). *Fazedores*. Acesso em 04 de Abril de 2016, disponível em fazedores: <http://blog.fazedores.com/arduino-conheca-esta-plataforma-de-hardware-livre-e-suas-aplicacoes/>

- 15 Macedo, Z. S., & Valerio, M. E. (Abril de 2004). Acesso em 14 de junho de 2016, disponível em Física Experimental: <https://docs.google.com/document/d/17jTAaVXMXVQqsfeaQqxAY4h3rZtu47sZbkrt-s5F2wjE/edit>

- 16 MGPesca. (2016). *mspesca*. Acesso em 03 de 04 de 2016, disponível em MGPesca Sua loja de pesca online: <http://www.mgpesca.com.br/carretilha-marine-sports-caster-400-400l-3bil-pr-340-340075.htm>

- 17 MPA/MMA. (13 de 06 de 2012). Instrução normativa Interministerial MPA/MMA Nº 9 .

- 18 O'Brien, /. (22 de Dezembro de 2008). *IBM*. Acesso em 04 de Abril de 2016, disponível em IBM developerWorks: <http://www.ibm.com/developerworks/br/library/os-arduino1/index.html>

- 19 Olo, A. (26 de 11 de 2014). *Dicas De Pesca*. Acesso em 05 de junho de 2016, disponível em Pesca & Companhia: <http://revistapescaecompanhia.com.br/dicas-de-pesca/peixesdecouroempesqueiros>

- 20 Pesca Amadora. (12 de 01 de 2016). *Pesca Amadora*. Acesso em 03 de Abril de 2016, disponível em pescamadora: <http://www.pescamadora.com.br/carretilhas/>

- 21 Sabedoriaeletronica. (23 de Fevereiro de 2014). *tipos de sensores*. Acesso em 5 de Abril de 2016, disponível em Sabedoriaeletronica: <http://www.sabereletronica.com.br/artigos/1532-todos-os-tipos-de-sensores>

- 22 Santos, J. F. (27 de Fevereiro de 2011). *Efeito Hall*. Acesso em 05 de Abril de 2016, disponível em Inape: <http://www.inape.org.br/colunas/fisica-conceito-historia/efeito-hall>

- 23 Silveira, A. (09 de 12 de 2013). *Pesca esportiva e Desenvolvimento*. Acesso em 28 de Março de 2016, disponível em A Última Arca de Noé:
<http://www.aultimaarcadenoe.com.br/pesca-esportiva-2/>
- 24 Silveira, J. A. (2013). *Arduino*. Acesso em 18 de Abril de 2016, disponível em ordemnatural: <http://www.ordemnatural.com.br/pdf-files/expArduino-caps1e2.pdf>
- 25 Souza, F. (29 de Novembro de 2013). *Arduino uno*. Acesso em 20 de Abril de 2016, disponível em embarcados: <http://www.embarcados.com.br/arduino-uno/>
- 26 Tanikawa, E. S. (2016). Acesso em 03 de Abril de 2016, disponível em sakaitanikawa:
<http://sakaitanikawa.tripod.com/Historia.htm>
- 27 Thomsen, A. (02 de Setembro de 2014). *FELIPEFLOP*. Acesso em 18 de Abril de 2016, disponível em Felipeflop: <http://blog.filipeflop.com/arduino/tipos-de-arduino-qual-comprar.html>
- 28 Thosen, A. (05 de 05 de 2014). *SENSOR HALL KY-003 COM SERVO TOWER PRO SG-5010*. Acesso em Abril de 10 de 2016, disponível em blog.felipeflop:
<http://blog.filipeflop.com/sensores/sensor-hall-servo-tower-pro-sg5010.html>
- 29 UFRGS. (2003). *Instituto de física UFRGS*. Acesso em 05 de Abril de 2016, disponível em ufrgs: <http://www.if.ufrgs.br/tex/fis142/mod08/>

APÊNCIDE A- CÓDIGO FONTE DA CARRETILHA INSTRUMENTADA NA PLATAFORMA ARDUINO

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,4,5,6,7);
#include <EEPROM.h>

// Porta ligada ao pino SINAL do sensor
int pinosinal = 1; //Sensor 1 da direita
int pinosinal2 = 3; //Sensor 2 esquerda
// Armazena informações sobre a leitura do sensor
int contador = 0; //variavel guarda quantidade de voltas dada no sentido horário
float voltas = 0; // variavel que recebe o calculo da quant
float voltasC = 0; // variavel que recebe o calculo da valor de linha presente no carretel no
cadastramento;
float y=0; // variavel que guarda a bitola de linha que esta sendo utilizada
float z=0; // variavel que guarda a quantidade de linha, em metros, que é referente a
determinada volta;
float total =0; // variavel total do sistema;
float totallinha = 0; // variavel do medição
int antihorario = 0; // variável de comparação sentido anti-horário
int horario= 0; // variável de comparação sentido horário
int bloqs1 = 0; // variável de bloqueio do sensor 1
int bloqs2 = 0; // variável de bloqueio sensor 2
int contador2 =0; // contador da medição
int VoltasTabela = 0; // variável de comparação na tabela
int totalVoltas = 0; // total voltas
int menu = 0; // variavel menu
int keypad_pin = A0;
int keypad_value = 0;
int keypad_value_old = 0;
char btn_push;
```

```

byte mainMenuPage = 1;
byte mainMenuPageOld = 1;
byte mainMenuTotal = 4;
int menuC = 1;

```

```

////////////////////

```

```

/*EEPROM estrutura

```

```

Endereco - Tipo - Dado
0    - INT  - Contador
2    - Float - Total
6    - Float - Y

```

```

PARA GRAVAR

```

```

EEPROM.put( 0, contador );
EEPROM.put( 2, total );
EEPROM.put( 6, y );

```

```

PARA LER

```

```

EEPROM.get( 0, contador );
EEPROM.get( 2, total );
EEPROM.get( 6, y );

```

```

*/

```

```

////////////////////

```

```

void setup()

```

```

{
    lcd.begin(16,2); //Initialize a 2x16 type LCD
    pinMode(pinosinal, INPUT);
    pinMode(pinosinal2, INPUT);
    MainMenuDisplay();
    delay(1000);

```

```

if (EEPROM.read(0) == 255){

```

```

EEPROM.put( 0, contador );
EEPROM.put( 2, total );
EEPROM.put( 6, y );
}
else{
    EEPROM.get( 0, contador );
    EEPROM.get( 2, total );
    EEPROM.get( 6, y );
    //totallinha = total;

}
}

void loop()
{
    btn_push = ReadKeypad();

    MainMenuBtn();

    if(btn_push == 'S')// ENTRA NO MENU SELECIONADO
    {
        WaitBtnRelease();
        switch (mainMenuPage)
        {
            case 1:
                CadastramentoPadrao();
                break;
            case 2:
                Medicao();
                break;
            case 3:
                MedicaoProfundidade();
                break;

```

```

        case 4:
            Ajuste();
            break;
        }
        MainMenuDisplay();
        WaitBtnRelease();
    }
    delay(10);
} //----- End of loop() loop -----

```

```

void MedicaoProfundidade()
{
    menu = 3; //MENU PRODUNDIDADE
    contador2 = 0;
    voltas = 0;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Profundidade: ");
    lcd.setCursor(0,1);
    lcd.print("    m");
    while(ReadKeypad() != 'L')
    {
        contalinhaMedicao();
    }
}

```

```

void CadastramentoPadrao()
{
    menu = 1; // MENU CADASTRAMENTO
    reset ();
    lcd.clear();
    lcd.setCursor(0,0);

```



```

lcd.print("Escolha linha");
seleciona(menuC);
while(ReadKeypad() != 'L')
{
if (ReadKeypad() == 'D'){
    menuC++;
    if (menuC > 3)
        menuC = 1;
    seleciona(menuC);
    Serial.print(y);
    delay(300);
}
if (ReadKeypad() == 'U'){
    menuC--;
    if (menuC < 1)
        menuC = 3;
    seleciona(menuC);
    delay(300);
}
if (ReadKeypad() == 'S'){
    EEPROM.put( 6, y );
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("LTotal: ");
    lcd.setCursor(0,1);
    lcd.print("    m");
while(ReadKeypad() != 'L')
{
    contalinhaCadastramento();
}
totalVoltas = contador;
total = voltasC;

```

```

EEPROM.put( 0, contador );
EEPROM.put( 2, total );
//total = totallinha;
voltasC = 0;
}
}
}

void Ajuste()
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Aperte R p/setar");
  lcd.setCursor(0,1);
  lcd.print("total para: ");
  lcd.setCursor(12,1);
  lcd.print(totallinha);
  while(ReadKeypad()!='L')
  {
    if(ReadKeypad()=='R')
    {
      total = totallinha;
      EEPROM.put( 2, total );
      voltas= 0;
      contador = contador +contador2;
      contador2 = 0;
    }
  }
}

void Medicao(){
  menu = 2; // ATUALIZAR MEDICAO

```

```

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("LTotal:    m");
    lcd.setCursor(8,0);
    lcd.print(total);
    lcd.setCursor(0,1);
    lcd.print("LSolta:    m");
    lcd.setCursor(8,1);
    lcd.print(voltas);
    while(ReadKeypad() != 'L')
    {
        contalinhaMedicao();
    }
}

void MainMenuDisplay()
{
    lcd.clear();
    lcd.setCursor(0,0);
    switch (mainMenuPage)
    {
        case 1:
            lcd.print("Cadastramento ");
            break;
        case 2:
            lcd.print(" Medicao");
            break;
        case 3:
            lcd.print("Profundidade ");
            break;
        case 4:
            lcd.print(" Ajuste");

```

```

        break;
    }
}

void MainMenuBtn()
{
    WaitBtnRelease();
    if(btn_push == 'U')
    {
        mainMenuPage++;
        if(mainMenuPage > mainMenuTotal)
            mainMenuPage = 1;
    }
    else if(btn_push == 'D')
    {
        mainMenuPage--;
        if(mainMenuPage == 0)
            mainMenuPage = mainMenuTotal;
    }

    if(mainMenuPage != mainMenuPageOld) // apenas atualiza o display quando a página for
    mudada
    {
        MainMenuDisplay();
        mainMenuPageOld = mainMenuPage;
    }
}

char ReadKeypad()
{
    keypad_value = analogRead(keypad_pin);
    if(keypad_value < 100)
        return 'R';
}

```

```

else if(keypad_value < 200)
    return 'U';
else if(keypad_value < 400)
    return 'D';
else if(keypad_value < 600)
    return 'L';
else if(keypad_value < 800)
    return 'S';
else
    return 'N';
}

void WaitBtnRelease()
{ while( analogRead(keypad_pin) < 800)}

void atdisp(){
    switch (menu) {
        case 1: //CADASTRAMENTO
            lcd.setCursor(0,1);
            lcd.print("  ");
            lcd.setCursor(0,1);
            if (voltasC <= 0){
                voltasC = 0;
                contador = 0;
            }
            lcd.print(voltasC);
            break;
        case 2: //MEDICAO
            if (voltas < 0){
                voltas = 0;
                contador2 = 0;
            }
    }
}

```

```

    if (voltas > total ){
        voltas = total;
        contador2 = -contador;
    }
    totallinha = total - voltas;
    lcd.setCursor(8,0);
    lcd.print("  ");
    lcd.setCursor(8,0);
    lcd.print(totallinha);
    lcd.setCursor(8,1);
    lcd.print ("  ");
    lcd.setCursor(8,1);
    lcd.print(voltas);
    break;
case 3: //PROFUNDIDADE
    lcd.setCursor(0,1);
    lcd.print("  ");
    lcd.setCursor(0,1);
    if (voltas < 0){
        voltas = 0;
        contador2 = 0;
    }
    lcd.print(voltas);
    break;
}
}

void seleciona(int x){
    lcd.setCursor(0,1);
    if (x == 1){
        lcd.print("(0.40)0.45 0.50 ");
    }
}

```

```

    y = 0.40;
}
if (x == 2){
    lcd.print(" 0.40(0.45)0.50 ");
    y = 0.45;
}
if (x == 3){
    lcd.print(" 0.40 0.45(0.50)");
    y = 0.50;
}
}

void contalinhaMedicao(){
    if (!digitalRead(pinosinal)){
        if (bloqs1 == 0){
            bloqs1 = 1;
            if (antihorario == 1){
                antihorario = 0;
                contador2 --;
                VoltasTabela = contador + contador2;
                MedicaoVolta();
                voltas = voltas + z;
                atdisp();
            }
            else
                horario = 1;
        }
    }
    else bloqs1 = 0;

    if (!digitalRead(pinosinal2)){
        if (bloqs2 == 0){

```

```

bloqs2 = 1;
if (horario == 1){
    horario = 0;
    contador2 ++;
    VoltasTabela = contador + contador2;
    MedicaoVolta();
    voltas = voltas - z;
    atdisp();
}
else{
    antihorario = 1;
}
}
}
else bloqs2 = 0;
}

```

```

void contalinhaCadastramento(){
if (!digitalRead(pinosinal)){
    if (bloqs1 == 0){
        bloqs1 = 1;
        if (antihorario == 1){
            antihorario = 0;
            contador --;
            VoltasTabela = contador;
            MedicaoVolta();
            voltasC = voltasC - z;
            atdisp();
        }
        else
            horario = 1;
    }
}
}

```



```

}
else bloqs1 = 0;
if (!digitalRead(pinosinal2)){
  if (bloqs2 == 0){
    bloqs2 = 1;
    if (horario == 1){
      horario = 0;
      contador ++;
      VoltasTabela = contador;
      MedicaoVolta();
      voltasC = voltasC + z;
      atdisp();
    }
  }
  else{
    antihorario = 1;
  }
}
}
else bloqs2 = 0;
}

```

void reset() // função responsável por zerar todas as variáveis quando cadastramento de nova linha for escolhido

```

{
  contador = 0;
  voltas = 0;
  voltasC = 0;
  y=0;
  z=0;
  total =0;
  totallinha = 0;
  contador2 =0;

```

```

VoltasTabela = 0;
totalVoltas = 0;
}

```

```

void MedicaoVolta( )
{
    if( y== 0.40)
    {
        if( VoltasTabela >= 1 & VoltasTabela <= 25)
            z= 0.0445;
        if( VoltasTabela >= 26 & VoltasTabela <= 52)
            z= 0.0460;
        if( VoltasTabela >= 53 & VoltasTabela <= 72)
            z= 0.0473;
        if( VoltasTabela >= 73 & VoltasTabela <= 97)
            z= 0.048;
        if( VoltasTabela >= 98 & VoltasTabela <= 120)
            z= 0.0482;
        if( VoltasTabela >= 121 & VoltasTabela <= 144)
            z= 0.0488;
        if( VoltasTabela >= 145 & VoltasTabela <= 170)
            z= 0.0494;
        if( VoltasTabela >= 171 & VoltasTabela <= 196)
            z= 0.0499;
        if( VoltasTabela >= 197 & VoltasTabela <= 218)
            z= 0.0515;
        if( VoltasTabela >= 219 & VoltasTabela <= 241)
            z= 0.0523;
        if( VoltasTabela >= 242 & VoltasTabela <= 266)
            z= 0.0531;
        if( VoltasTabela >= 267 & VoltasTabela <= 292)
            z= 0.0539;
    }
}

```

```
if( VoltasTabela >= 293 & VoltasTabela <= 316)
z= 0.0546;
if( VoltasTabela >= 317 & VoltasTabela <= 343)
z= 0.0552;
if( VoltasTabela >= 344 & VoltasTabela <= 364)
z= 0.0559;
if( VoltasTabela >= 365 & VoltasTabela <= 390)
z= 0.0566;
if( VoltasTabela >= 391 & VoltasTabela <= 414)
z= 0.0575;
if( VoltasTabela >= 415 & VoltasTabela <= 440)
z= 0.0584;
    if( VoltasTabela >= 441 & VoltasTabela <= 464)
z= 0.0593;
if( VoltasTabela >= 465 & VoltasTabela <= 490)
z= 0.0601;
if( VoltasTabela >= 491 & VoltasTabela <= 512)
z= 0.0609;
if( VoltasTabela >= 513 & VoltasTabela <= 536)
z= 0.0626;
if( VoltasTabela >= 537 & VoltasTabela <= 562)
z= 0.0633;
if( VoltasTabela >= 563 & VoltasTabela <= 584)
z= 0.0639;
if( VoltasTabela >= 585 & VoltasTabela <= 610)
z= 0.0647;
if( VoltasTabela >= 611 & VoltasTabela <= 634)
z= 0.0654;
if( VoltasTabela >= 635 & VoltasTabela <= 660)
z= 0.0662;
    if( VoltasTabela >= 661 & VoltasTabela <= 682)
z= 0.0669;
```

```
if( VoltasTabela >= 683 & VoltasTabela <= 708)
z= 0.0687;
if( VoltasTabela >= 709 & VoltasTabela <= 732)
z= 0.0705;
if( VoltasTabela >= 733 & VoltasTabela <= 760)
z= 0.0714;
if( VoltasTabela >= 761 & VoltasTabela <= 784)
z= 0.0721;
if( VoltasTabela >= 785 & VoltasTabela <= 809)
z= 0.0729;
if( VoltasTabela >= 810 & VoltasTabela <= 832)
z= 0.0737;
if( VoltasTabela >= 833 & VoltasTabela <= 860)
z= 0.0744;
if( VoltasTabela >= 861 & VoltasTabela <= 882)
z= 0.0752;
if( VoltasTabela >= 883 & VoltasTabela <= 908)
z= 0.0761;
if( VoltasTabela >= 909 & VoltasTabela <= 933)
z= 0.0768;
if( VoltasTabela >= 934 & VoltasTabela <= 954)
z= 0.0776;
if( VoltasTabela >= 955 & VoltasTabela <= 980)
z= 0.0785;
if( VoltasTabela >= 981 & VoltasTabela <= 1006)
z= 0.0794;
if( VoltasTabela >= 1007 & VoltasTabela <= 1033)
z= 0.0803;
if( VoltasTabela >= 1034 & VoltasTabela <= 1057)
z= 0.0811;
if( VoltasTabela >= 1058 & VoltasTabela <= 1082)
z= 0.0820;
```

```
if( VoltasTabela >= 1083 & VoltasTabela <= 1108)
z= 0.0828;
if( VoltasTabela >= 1109 & VoltasTabela <= 1131)
z= 0.0836;
if( VoltasTabela >= 1132 & VoltasTabela <= 1156)
z= 0.0845;
if( VoltasTabela >= 1157 & VoltasTabela <= 1182)
z= 0.0853;
if( VoltasTabela >= 1183 & VoltasTabela <= 1209)
z= 0.0860;
if( VoltasTabela >= 1210 & VoltasTabela <= 1233)
z= 0.0868;
if( VoltasTabela >= 1234 & VoltasTabela <= 1258)
z= 0.0875;
if( VoltasTabela >= 1259 & VoltasTabela <= 1284)
z= 0.0882;
if( VoltasTabela >= 1285 & VoltasTabela <= 1308)
z= 0.0890;
if( VoltasTabela >= 1309 & VoltasTabela <= 1333)
z= 0.0898;
if( VoltasTabela >= 1334 & VoltasTabela <= 1357)
z= 0.0904;
if( VoltasTabela >= 1358 & VoltasTabela <= 1383)
z= 0.0911;
if( VoltasTabela >= 1384 & VoltasTabela <= 1408)
z= 0.0919;
if( VoltasTabela >= 1409 & VoltasTabela <= 1432)
z= 0.0927;
if( VoltasTabela >= 1433 & VoltasTabela <= 1455)
z= 0.0934;
if( VoltasTabela >= 1456 & VoltasTabela <= 1480)
z= 0.0942;
```

```
if( VoltasTabela >= 1481 & VoltasTabela <= 1506)
z= 0.0952;
if( VoltasTabela >= 1507 & VoltasTabela <= 1530)
z= 0.0960;
if( VoltasTabela >= 1531 & VoltasTabela <= 1555)
z= 0.0967;
if( VoltasTabela >= 1556 & VoltasTabela <= 1581)
z= 0.0970;
if( VoltasTabela >= 1582 & VoltasTabela <= 1608)
z= 0.0974;
if( VoltasTabela >= 1609 & VoltasTabela <= 1633)
z= 0.0982;
if( VoltasTabela >= 1634 & VoltasTabela <= 1657)
z= 0.0991;
if( VoltasTabela >= 1658 & VoltasTabela <= 1683)
z= 0.0999;
if( VoltasTabela >= 1684 & VoltasTabela <= 1708)
z= 0.1006;
if( VoltasTabela >= 1709 & VoltasTabela <= 1734)
z= 0.1014;
if( VoltasTabela >= 1735 & VoltasTabela <= 1759)
z= 0.1027;
if( VoltasTabela >= 1760 & VoltasTabela <= 1783)
z= 0.1036;
if( VoltasTabela >= 1784 & VoltasTabela <= 1808)
z= 0.1047;
if( VoltasTabela >= 1809 & VoltasTabela <= 1832)
z= 0.1055;
if( VoltasTabela >= 1833 & VoltasTabela <= 1858)
z= 0.1059;
if( VoltasTabela >= 1859 & VoltasTabela <= 1883)
z= 0.1065;
```

```
if( VoltasTabela >= 1884 & VoltasTabela <= 1907)
z= 0.1078;
if( VoltasTabela >= 1908 & VoltasTabela <= 1933)
z= 0.1088;
if( VoltasTabela >= 1934 & VoltasTabela <= 1958)
z= 0.1095;
if( VoltasTabela >= 1959 & VoltasTabela <= 1981)
z= 0.1103;
if( VoltasTabela >= 1982 & VoltasTabela <= 2006)
z= 0.1112;
if( VoltasTabela >= 2007 & VoltasTabela <= 2030)
z= 0.1119;
if( VoltasTabela >= 2031 & VoltasTabela <= 2053)
z= 0.1125;
if( VoltasTabela >= 2054 & VoltasTabela <= 2078)
z= 0.1130 ;
if( VoltasTabela >= 2079 & VoltasTabela <= 2104)
z= 0.1143;
if( VoltasTabela >= 2105 & VoltasTabela <= 2128)
z= 0.1151;
if( VoltasTabela >= 2129 & VoltasTabela <= 2153)
z= 0.1160;
if( VoltasTabela >= 2154 & VoltasTabela <= 2176)
z= 0.1169;
if( VoltasTabela >= 2177 & VoltasTabela <=2200 )
z= 0.1178;
if( VoltasTabela >= 2201 & VoltasTabela <= 2225)
z= 0.1189;
if( VoltasTabela >= 2226 & VoltasTabela <= 2249)
z= 0.1195;
if( VoltasTabela >= 2250 & VoltasTabela <= 2273)
z= 0.1204;
```

```

if( VoltasTabela >= 2274 & VoltasTabela <= 2298)
z= 0.1212;
if( VoltasTabela >= 2299 & VoltasTabela <=2324 )
z= 0.1223;
if( VoltasTabela >= 2325 & VoltasTabela <= 2348)
z= 0.1230;
if( VoltasTabela >= 2349 & VoltasTabela <= 2373)
z= 0.1238;
if( VoltasTabela >= 2374 & VoltasTabela <= 2398)
z= 0.1245;
if( VoltasTabela >= 2399 & VoltasTabela <= 2422)
z= 0.1250;
if( VoltasTabela >= 2423 & VoltasTabela <= 2448)
z= 0.1266;
if( VoltasTabela >= 2449 & VoltasTabela <= 2472)
z= 0.1270;
if( VoltasTabela >= 2473 & VoltasTabela <= 2497 )
z= 0.1283;
if( VoltasTabela >= 2498 )
z= 0.1291;

}

if( y== 0.45)
{
if( VoltasTabela >= 1 & VoltasTabela <= 28)
z= 0.0446;
if( VoltasTabela >= 29 & VoltasTabela <= 52)
z= 0.0454;
if( VoltasTabela >= 53 & VoltasTabela <= 77)
z= 0.0463;
if( VoltasTabela >= 78 & VoltasTabela <= 102)
z= 0.0484;

```



```
if( VoltasTabela >= 103 & VoltasTabela <= 126)
z= 0.0493;
if( VoltasTabela >= 127 & VoltasTabela <= 149)
z= 0.0498;
if( VoltasTabela >= 150 & VoltasTabela <= 172)
z= 0.0512;
if( VoltasTabela >= 173 & VoltasTabela <= 200)
z= 0.0533;
if( VoltasTabela >= 201 & VoltasTabela <= 222)
z= 0.0558;
    if( VoltasTabela >= 223 & VoltasTabela <= 247)
z= 0.0576;
if( VoltasTabela >= 248 & VoltasTabela <= 270)
z= 0.0584;
if( VoltasTabela >= 271 & VoltasTabela <= 296)
z= 0.0596;
if( VoltasTabela >= 297 & VoltasTabela <= 321)
z= 0.0606;
if( VoltasTabela >= 322 & VoltasTabela <= 346)
z= 0.0612;
if( VoltasTabela >= 347 & VoltasTabela <= 370)
z= 0.0618;
if( VoltasTabela >= 371 & VoltasTabela <= 396)
z= 0.0627;
if( VoltasTabela >= 397 & VoltasTabela <= 420)
z= 0.0640;
if( VoltasTabela >= 421 & VoltasTabela <= 444)
z= 0.0655;
    if( VoltasTabela >= 445 & VoltasTabela <= 470)
z= 0.0666;
if( VoltasTabela >= 471 & VoltasTabela <= 493)
z= 0.0679;
```

```
if( VoltasTabela >= 494 & VoltasTabela <= 518)
z= 0.0688;
if( VoltasTabela >= 519 & VoltasTabela <= 542)
z= 0.0699;
if( VoltasTabela >= 543 & VoltasTabela <= 567)
z= 0.0712;
if( VoltasTabela >= 568 & VoltasTabela <= 592)
z= 0.0726;
if( VoltasTabela >= 593 & VoltasTabela <= 617)
z= 0.0740;
if( VoltasTabela >= 618 & VoltasTabela <= 641)
z= 0.0752;
if( VoltasTabela >= 642 & VoltasTabela <= 666)
z= 0.0771;
if( VoltasTabela >= 667 & VoltasTabela <= 689)
z= 0.0783;
if( VoltasTabela >= 690 & VoltasTabela <= 714)
z= 0.0792;
if( VoltasTabela >= 715 & VoltasTabela <= 739)
z= 0.0805;
if( VoltasTabela >= 740 & VoltasTabela <= 762)
z= 0.0817;
if( VoltasTabela >= 763 & VoltasTabela <= 786)
z= 0.0830;
if( VoltasTabela >= 787 & VoltasTabela <= 814)
z= 0.0843;
if( VoltasTabela >= 815 & VoltasTabela <= 836)
z= 0.0856;
if( VoltasTabela >= 837 & VoltasTabela <= 860)
z= 0.0865;
if( VoltasTabela >= 861 & VoltasTabela <= 884)
z= 0.0876;
```

```
if( VoltasTabela >= 885 & VoltasTabela <= 912)
z= 0.0887;
if( VoltasTabela >= 913 & VoltasTabela <= 934)
z= 0.0896;
if( VoltasTabela >= 935 & VoltasTabela <= 962)
z= 0.0905;
if( VoltasTabela >= 963 & VoltasTabela <= 984)
z= 0.0916;
if( VoltasTabela >= 985 & VoltasTabela <= 1010)
z= 0.0927;
if( VoltasTabela >= 1011 & VoltasTabela <= 1032)
z= 0.0939;
if( VoltasTabela >= 1033 & VoltasTabela <= 1060)
z= 0.0948;
if( VoltasTabela >= 1061 & VoltasTabela <= 1082)
z= 0.0960;
if( VoltasTabela >= 1083 & VoltasTabela <= 1108)
z= 0.0974;
if( VoltasTabela >= 1109 & VoltasTabela <= 1122)
z= 0.0987;
if( VoltasTabela >= 1129 & VoltasTabela <= 1150)
z= 0.1001;
if( VoltasTabela >= 1151 & VoltasTabela <= 1170)
z= 0.1014;
if( VoltasTabela >= 1171 & VoltasTabela <= 1196)
z= 0.1025;
if( VoltasTabela >= 1197 & VoltasTabela <= 1220)
z= 0.1038;
if( VoltasTabela >= 1221 & VoltasTabela <= 1247)
z= 0.1047;
if( VoltasTabela >= 1248 & VoltasTabela <= 1273)
z= 0.1057;
```

```
if( VoltasTabela >= 1274 & VoltasTabela <= 1298)
z= 0.1069;
if( VoltasTabela >= 1299 & VoltasTabela <= 1323)
z= 0.1080;
if( VoltasTabela >= 1324 & VoltasTabela <= 1348)
z= 0.1093;
if( VoltasTabela >= 1349 & VoltasTabela <= 1371)
z= 0.1102;
if( VoltasTabela >= 1372 & VoltasTabela <= 1396)
z= 0.1116;
if( VoltasTabela >= 1397 & VoltasTabela <= 1422)
z= 0.1126;
if( VoltasTabela >= 1423 & VoltasTabela <= 1448)
z= 0.1136;
if( VoltasTabela >= 1449 & VoltasTabela <= 1475)
z= 0.1147;
if( VoltasTabela >= 1476 & VoltasTabela <= 1501)
z= 0.1159;
if( VoltasTabela >= 1502 & VoltasTabela <= 1528)
z= 0.1172;
if( VoltasTabela >= 1529 & VoltasTabela <= 1554)
z= 0.1183;
if( VoltasTabela >= 1555 & VoltasTabela <= 1581)
z= 0.1194;
if( VoltasTabela >= 1582 & VoltasTabela <= 1607)
z= 0.1201;
if( VoltasTabela >= 1608 & VoltasTabela <= 1633)
z= 0.1211;
if( VoltasTabela >= 1634 & VoltasTabela <= 1660)
z= 0.1222;
if( VoltasTabela >= 1661 & VoltasTabela <= 1687)
z= 0.1237;
```

```

if( VoltasTabela >= 1688 & VoltasTabela <= 1713)
z= 0.1245;
if( VoltasTabela >= 1714 & VoltasTabela <= 1740)
z= 0.1254;
if( VoltasTabela >= 1741 & VoltasTabela <= 1767)
z= 0.1261;
if( VoltasTabela >= 1768)
z= 0.1271;

}
if( y== 0.50)
{
if( VoltasTabela >= 1 & VoltasTabela <= 25)
z= 0.0463;
if( VoltasTabela >= 26 & VoltasTabela <= 47)
z= 0.0477;
if( VoltasTabela >= 48 & VoltasTabela <= 70)
z= 0.0484;
if( VoltasTabela >= 71 & VoltasTabela <= 96)
z= 0.0492;
if( VoltasTabela >= 97 & VoltasTabela <= 117)
z= 0.0503;
if( VoltasTabela >= 118 & VoltasTabela <= 144)
z= 0.0516;
if( VoltasTabela >= 145 & VoltasTabela <= 168)
z= 0.0525;
if( VoltasTabela >= 169 & VoltasTabela <= 195)
z= 0.0541;
if( VoltasTabela >= 196 & VoltasTabela <= 217)
z= 0.0555;
if( VoltasTabela >= 218 & VoltasTabela <= 241)
z= 0.0569;

```

```
if( VoltasTabela >= 242 & VoltasTabela <= 265)
z= 0.0581;
if( VoltasTabela >= 266 & VoltasTabela <= 288)
z= 0.0597;
if( VoltasTabela >= 289 & VoltasTabela <= 314)
z= 0.0613;
if( VoltasTabela >= 315 & VoltasTabela <= 338)
z= 0.0621;
if( VoltasTabela >= 339 & VoltasTabela <= 364)
z= 0.0637;
if( VoltasTabela >= 365 & VoltasTabela <= 388)
z= 0.0650;
if( VoltasTabela >= 389 & VoltasTabela <= 414)
z= 0.0665;
if( VoltasTabela >= 415 & VoltasTabela <= 438)
z= 0.0683;
if( VoltasTabela >= 439 & VoltasTabela <= 462)
z= 0.0696;
if( VoltasTabela >= 463 & VoltasTabela <= 484)
z= 0.0708;
if( VoltasTabela >= 485 & VoltasTabela <= 510)
z= 0.0713;
if( VoltasTabela >= 511 & VoltasTabela <= 535)
z= 0.0724;
if( VoltasTabela >= 536 & VoltasTabela <= 558)
z= 0.0739;
if( VoltasTabela >= 559 & VoltasTabela <= 584)
z= 0.0757;
if( VoltasTabela >= 585 & VoltasTabela <= 608)
z= 0.0774;
if( VoltasTabela >= 609 & VoltasTabela <= 631)
z= 0.0781;
```

```
if( VoltasTabela >= 632 & VoltasTabela <= 657)
z= 0.0786;
if( VoltasTabela >= 658 & VoltasTabela <= 681)
z= 0.0793;
if( VoltasTabela >= 682 & VoltasTabela <= 705)
z= 0.0804;
if( VoltasTabela >= 706 & VoltasTabela <= 727)
z= 0.0818;
if( VoltasTabela >= 728 & VoltasTabela <= 754)
z= 0.0827;
if( VoltasTabela >= 755 & VoltasTabela <= 777)
z= 0.0841;
if( VoltasTabela >= 778 & VoltasTabela <= 804)
z= 0.0854;
if( VoltasTabela >= 805 & VoltasTabela <= 827)
z= 0.0871;
if( VoltasTabela >= 828 & VoltasTabela <= 854)
z= 0.0879;
if( VoltasTabela >= 855 & VoltasTabela <= 875)
z= 0.0887;
if( VoltasTabela >= 876 & VoltasTabela <= 900)
z= 0.0896;
if( VoltasTabela >= 901 & VoltasTabela <= 923)
z= 0.0909;
if( VoltasTabela >= 924 & VoltasTabela <= 947)
z= 0.0923;
if( VoltasTabela >= 948 & VoltasTabela <= 973)
z= 0.0932;
if( VoltasTabela >= 974 & VoltasTabela <= 998)
z= 0.0944;
if( VoltasTabela >= 999 & VoltasTabela <= 1020)
z= 0.0952;
```

```
if( VoltasTabela >= 1021 & VoltasTabela <= 1046)
z= 0.0969;
if( VoltasTabela >= 1047 & VoltasTabela <= 1068)
z= 0.0986;
if( VoltasTabela >= 1069 & VoltasTabela <= 1094)
z= 0.1000;
if( VoltasTabela >= 1095 & VoltasTabela <= 1118)
z= 0.1011;
if( VoltasTabela >= 1119 & VoltasTabela <= 1142)
z= 0.1024;
if( VoltasTabela >= 1143 & VoltasTabela <= 1167)
z= 0.1037;
if( VoltasTabela >= 1168 & VoltasTabela <= 1192)
z= 0.1059;
if( VoltasTabela >= 1193 & VoltasTabela <= 1218)
z= 0.1078;
if( VoltasTabela >= 1219 & VoltasTabela <= 1243)
z= 0.1094;
if( VoltasTabela >= 1244 & VoltasTabela <= 1268)
z= 0.1110;
if( VoltasTabela >= 1269 & VoltasTabela <= 1292)
z= 0.1123;
if( VoltasTabela >= 1293 & VoltasTabela <= 1314)
z= 0.1134;
if( VoltasTabela >= 1315)
z= 0.1144;
}
}
```


APÊNDICE B- QUADROS DE MEDIÇÃO DAS BITOLAS DE LINHA 0.40, 0.45 E 0.50

Quadro medição linha 0,40		
Volta	Camada	Média das 3 medições
1~25	1	0,0445
26~52	2	0,0460
53~72	3	0,0473
73~97	4	0,0480
98~120	5	0,0482
121~144	6	0,0488
145~170	7	0,0494
171~196	8	0,0499
197~218	9	0,0515
219~241	10	0,0523
242~266	11	0,0531
267~292	12	0,0539
293~316	13	0,0546
317~343	14	0,0552
344~364	15	0,0559
365~390	16	0,0566
391~414	17	0,0575
415~440	18	0,0584
441~464	19	0,0593
465~490	20	0,0601
491~512	21	0,0609
513~536	22	0,0626
537~562	23	0,0633
563~584	24	0,0639
585~610	25	0,0647
611~634	26	0,0654
635~660	27	0,0662
661~682	28	0,0669
683~708	29	0,0687
709~732	30	0,0705
733~760	31	0,0714
761~784	32	0,0721
785~809	33	0,0729
810~832	34	0,0737
833~860	35	0,0744
861~882	36	0,0752
883~908	37	0,0761
909~933	38	0,0768
934~954	39	0,0776
955~980	40	0,0785

981~1006	41	0,0794
1007~1033	42	0,0803
1034~1057	43	0,08011
1058~1082	44	0,0820
1083~1108	45	0,0828
1109~1131	46	0,0836
1132~1156	47	0,0845
1157~1182	48	0,0853
1183~1209	49	0,0860
1210~1233	50	0,0868
1234~1258	51	0,0875
1259~1284	52	0,0882
1285~1308	53	0,0890
1309~1333	54	0,0898
1334~1357	55	0,0904
1358~1383	56	0,0911
1384~1408	57	0,0919
1409~1432	58	0,0927
1433~1455	59	0,0934
1456~1480	60	0,0942
1481~1506	61	0,0952
1507~1530	62	0,0960
1531~1555	63	0,0967
1556~1581	64	0,0970
1582~1608	65	0,0974
1609~1633	66	0,0982
1634~1657	67	0,0991
1658~1683	68	0,0999
1684~1708	69	0,1006
1709~1734	70	0,1014
1735~1759	71	0,1027
1760~1783	72	0,1036
1784~1808	73	0,1047
1809~1832	74	0,1055
1833~1858	75	0,1059
1859~1883	76	0,1065
1884~1907	77	0,1078
1908~1933	78	0,1088
1934~1958	79	0,1095
1959~1981	80	0,1103
1982~2006	81	0,1112
2007~2030	82	0,1119
2031~2053	83	0,1125
2054~2078	84	0,1130
2079~2104	85	0,1143
2105~2128	86	0,1151

2129~2153	87	0,1160
2154~2176	88	0,1169
2177~2200	89	0,1178
2201~2225	90	0,1189
2226~2249	91	0,1195
2250~2273	92	0,1204
2274~2298	93	0,1212
2299~2324	94	0,1223
2325~2348	95	0,1230
2349~2373	96	0,1238
2374~2398	97	0,1245
2399~2422	98	0,1250
2423~2448	99	0,1266
2449~2472	100	0,1270
2473~2497	101	0,1283
>=2523	102	0,1291
Total de linha		190m

Quadro medição linha 0,45		
Volta	Camada	Média das 3 medições
1~28	1	0,0446
29~52	2	0,0454
53~77	3	0,0463
78~102	4	0,0484
103~126	5	0,0493
127~149	6	0,0498
150~172	7	0,0512
173~200	8	0,0533
201~222	9	0,0558
223~247	10	0,0576
248~270	11	0,0584
271~296	12	0,0596
297~321	13	0,0606
322~346	14	0,0612
347~370	15	0,0618
371~396	16	0,0627
397~420	17	0,0640
421~444	18	0,0655
445~470	19	0,0666
471~493	20	0,0679
494~518	21	0,0688
519~542	22	0,0699

543~567	23	0,0712
568~592	24	0,0726
593~617	25	0,0740
618~641	26	0,0752
642~666	27	0,0771
667~689	28	0,0783
690~714	29	0,0792
715~739	30	0,0805
740~762	31	0,0817
763~786	32	0,0830
787~814	33	0,0843
815~836	34	0,0856
837~860	35	0,0865
861~884	36	0,0876
885~912	37	0,0887
913~934	38	0,0896
935~962	39	0,0905
963~984	40	0,0916
985~1010	41	0,0927
1011~1032	42	0,0939
1033~1060	43	0,0948
1061~1082	44	0,0960
1083~1108	45	0,0974
1109~1122	46	0,0987
1123~1150	47	0,1001
1151~1170	48	0,1014
1171~1196	49	0,1025
1197~1220	50	0,1038
1221~1247	51	0,1047
1248~1273	52	0,1057
1274~1298	53	0,1069
1299~1323	54	0,1080
1324~1348	55	0,1093
1349~1371	56	0,1102
1372~1396	57	0,1116
1397~1422	58	0,1126
1423~1448	59	0,1136
1449~1475	60	0,1147
1476~1501	61	0,1159
1502~1528	62	0,1172
1529~1554	63	0,1183
1555~1581	64	0,1194
1582~1607	65	0,1201
1608~1633	66	0,1211
1634~1660	67	0,1222
1661~1687	68	0,1237

1688~1713	69	0,1245
1714~1740	70	0,1254
1741~1767	71	0,1261
>=1768	72	0,1271
Total de linha		135m

Quadro medição linha 0,50		
Volta	Camada	Média das 3 medições
1~25	1	0,0463
26~47	2	0,0477
48~70	3	0,0484
71~96	4	0,0492
97~117	5	0,0503
118~144	6	0,0516
145~168	7	0,0525
169~195	8	0,0541
196~217	9	0,0555
218~241	10	0,0569
242~265	11	0,0581
266~288	12	0,0597
289~314	13	0,0613
315~338	14	0,0621
339~364	15	0,0637
365~388	16	0,0650
389~414	17	0,0665
415~438	18	0,0683
439~462	19	0,0696
463~484	20	0,0708
485~510	21	0,0713
511~535	22	0,0724
536~558	23	0,0739
559~584	24	0,0757
585~608	25	0,0774
609~631	26	0,0781
632~657	27	0,0786
658~681	28	0,0793
682~705	29	0,0804
706~727	30	0,0814
728~754	31	0,0827
755~777	32	0,0841
778~804	33	0,0854

805~827	34	0,0871
828~854	35	0,0879
855~875	36	0,0887
876~900	37	0,0896
901~923	38	0,0909
924~947	39	0,0923
948~973	40	0,0932
974~998	41	0,0944
999~1020	42	0,0952
1021~1046	43	0,0969
1047~1068	44	0,0986
1069~1094	45	0,1000
1095~1118	46	0,1011
1119~1142	47	0,1024
1143~1167	48	0,1037
1168~1192	49	0,1059
1193~1218	50	0,1078
1219~1243	51	0,1094
1244~1268	52	0,1110
1269~1292	53	0,1123
1293~1314	54	0,1134
>=1315	55	0,1144
Total de linha		103 m